Windows 2000-Webserver – Empfohlene Vorgehensweisen für eine hohe Verfügbarkeit

(Engl. Originaltitel: Windows 2000 Web Server Best Practices for High Availability)

Betriebssystem

Von Tim Hodgkins

Microsoft Corporation

Veröffentlicht: Oktober 2001

Zusammenfassung

Zum Erreichen einer hohen Serververfügbarkeit auf einem Webserver ist eine Kombination aus optimal entworfenen und getesteten Anwendungen, umfassend getesteter Serverhardware sowie strenger Serverüberwachung und -verwaltung erforderlich. In diesem Artikel werden Strategien und Techniken untersucht, die Organisationen verwenden können, um eine Verfügbarkeit eines Einzelservers von 99,99 % mithilfe von Microsoft-Technologien zu erreichen.

Danksagungen

Katie Beers, Reza Baghai, Wael Bahaa-El-Din, Mario Garzia, Björn Levidow, Matthew Kerner, Ram Papatla, Michael Risse, Microsoft Corporation.

Einführung

Zum Erreichen einer hohen Serververfügbarkeit auf einem Webserver ist eine Kombination aus optimal entworfenen und getesteten Anwendungen, umfassend getesteter Serverhardware sowie strenger Serverüberwachung und -verwaltung erforderlich. Seit der Produkteinführung des Betriebssystems Windows 2000 hat Microsoft mit zehn externen, hoch entwickelten Dotcom-Unternehmen sowie fünf Microsoft-Gesellschaften zusammengearbeitet, um die Verfügbarkeit der jeweiligen Websites zu messen und zu verbessern. Die wichtigste Erkenntnis, die aus diesen wertvollen Beziehungen gewonnen werden konnte, bezieht sich auf Kenntnisse in Bezug auf die Auswirkungen betrieblicher Vorgehensweisen auf die Serververfügbarkeit.

In diesem Whitepaper werden Strategien und Techniken untersucht, die Organisationen verwenden können, um eine Verfügbarkeit eines Einzelservers von 99,99 % mithilfe von Microsoft-Technologien zu erreichen. Im ersten Abschnitt werden allgemeine Begriffe hinsichtlich Verfügbarkeit und Zuverlässigkeit definiert, die Beziehung zwischen den Variablen der Serververfügbarkeit beschrieben sowie Tools aufgelistet, die Microsoft erstellt hat, um Organisationen beim Messen der Serververfügbarkeit und -zuverlässigkeit zu unterstützen.

Im nächsten Abschnitt geht es um den ersten Schritt zum Erreichen einer Verfügbarkeit eines Einzelservers von 99,99 %. In diesem Zusammenhang werden empfohlene Vorgehensweisen beim Entwerfen und Entwickeln einer eigenen Webanwendung vorgestellt. Dieser Teil wurde in erster Linie für Webanwendungsentwickler verfasst. Für Webserveradministratoren ist dieser Abschnitt jedoch vor allem bei der Behandlung von Webserverproblemen hilfreich. Den Schwerpunkt dieser Passage bildet der Austausch empfohlener Vorgehensweisen zum Entwickeln hervorragender Komponenten und ASP-Prozesse (Active Server Page®) mit zusätzlicher Erläuterung der Datenbankkonnektivität.

Im dritten Abschnitt des Whitepaper wird der nächste Schritt zum Erreichen einer Verfügbarkeit eines Einzelservers von 99,99 % erläutert – das Entwickeln und Bereitstellen von Servern mit hoher Verfügbarkeit. Dieser Abschnitt ist auf Webadministratoren ausgelegt und befasst sich schwerpunktmäßig mit den Schritten, die ausgeführt werden können, um die Stabilität der Server vor und nach der Bereitstellung sicherzustellen. Mithilfe von umfassenden Tests und Leistungsoptimierungen der Anwendungs- und Hardwarekompatibilität wird sichergestellt, dass die Webserver eine hohe Verfügbarkeitsstufe aufrechterhalten können. Abschließend werden in diesem Abschnitt Technologien für Fehlertoleranz und Lastenausgleich erläutert.

Im letzten Abschnitt, der an Webserveradministratoren gerichtet ist, werden empfohlene Vorgehensweisen beschrieben, die eine Organisation zum Verwalten von Webservern verwenden kann. In diesem Abschnitt werden die wichtigsten Serververwaltungstools von Microsoft vorgestellt. Anschließend werden proaktive empfohlene Vorgehensweisen erläutert, die eine Organisation zum Minimieren der Serverausfallzeiten verwenden kann. Dies umfasst beispielsweise Tipps zum Debuggen sowie Hotfix- und Sicherheitsaktualisierungen.

Im gesamten Whitepaper werden Szenarien mit tatsächlich aufgetretenen Kundenproblemen zusammen mit Empfehlungen erläutert, die Ihre Organisation zum Identifizieren und Lösen dieser häufig auftretenden Probleme verwenden kann.

Verwenden Sie die unten aufgeführte Liste, um die zehn wichtigsten Empfehlungen kennen zu lernen, die Ihre Organisation nutzen kann, um die Zuverlässigkeit und Verfügbarkeit der Webserver zu verbessern:

- Wenn Sie die Anwendung in Visual Basic entwickeln, sollten Sie das Dienstprogramm VBCHKW2K verwenden, um die Richtigkeit der Kompilierungseinstellungen überprüfen zu können.
- Testen Sie die vorhandene Hardware und Software sorgfältig im Hinblick auf die Windows 2000-Kompatibilität.
- Verwenden Sie möglichst häufig gepoolte ASP-Seiten.
- Verwenden Sie möglichst häufig gepoolte COM-Komponenten.
- Verwenden Sie das Web Capacity Analysis-Tool und die HTTP-Überwachung, um Belastungstests der Anwendung durchzuführen.
- Dokumentieren und befolgen Sie einen Webserver-Bereitstellungsplan.
- Installieren und verwenden Sie das IIS 5 Recycle-Tool, um die Webserververfügbarkeit zu erhöhen.
- Dokumentieren und befolgen Sie einen Prozess zur Auswertung und Prioritäteneinteilung der Anwendbarkeit von freigegebenen Hotfixes und Sicherheitsaktualisierungen.
- Verwenden Sie HFCheck und QFECheck, um eine Betriebssystem-Standardinstallation auf allen Servern sicherzustellen.
- Wählen Sie ein Ziel für die Serververfügbarkeit aus, und verwenden Sie Microsoft-Tools, um den bisherigen Erfolg dieses Zieles nachverfolgen zu können.

Die hier erwähnten Probleme und Empfehlungen werden im Zusammenhang mit den Kundenszenarien und Codebeispielen im weiteren Verlauf dieses Whitepaper noch eingehender erläutert und können von Ihrer Organisation verwendet werden, um die Webserververfügbarkeit und -zuverlässigkeit zu verstehen und zu verbessern.

Anmerkung Der Schwerpunkt dieses Whitepapers liegt auf der Serververfügbarkeit im Gegensatz zur Verfügbarkeit einer Website oder eines anderen Dienstes. Bei allen in diesem Whitepaper erläuterten Tools und empfohlenen Vorgehensweisen steht das Messen und Verbessern der Verfügbarkeit von Einzelservern im Vordergrund. Das Hinzufügen von fehlertoleranten Konfigurationen und des Netzwerklastenausgleichs (Network Load Balancing oder NLB) auf einzelnen Servern mit hoher Verfügbarkeit und Zuverlässigkeit führt zu einer Website mit hoher Verfügbarkeit.

Auswählen eines Zieles für die Serververfügbarkeit

Terminologie zur Verfügbarkeit und Zuverlässigkeit

In diesem Abschnitt werden die in diesem Artikel im Zusammenhang mit Verfügbarkeit und Zuverlässigkeit verwendeten Begriffe definiert.

Zuverlässigkeit – die Wahrscheinlichkeit, dass das Produkt (System) die entsprechend vorgesehene Funktion in einem angegebenen Zeitraum ausführt, wenn der Betrieb unter normalen (oder festgelegten) Bedingungen erfolgt.[1]

In diesem Whitepaper wird die Zuverlässigkeit eines einzelnen Servers ermittelt, indem die durchschnittliche Zeit zwischen Ausfällen dieses Servers berechnet wird. Bei der durchschnittlichen Zeit zwischen Ausfällen, die im Allgemeinen auch als "durchschnittliche Zeit zwischen Neustarts" bezeichnet wird, handelt es sich um den Durchschnittswert zwischen den Ausfällen auf einem Server. Obwohl die Neustarts nicht die Folge eines Ausfalls sind, führen alle Neustarts zu Ausfallzeiten und können daher unter dem Gesichtspunkt der Zuverlässigkeit als Ausfall angesehen werden. Da Server problemlos mehrere Tage ohne Neustart ausgeführt werden können, wird die durchschnittliche Zeit zwischen Ausfällen in Tagen protokolliert.

Durchschnittliche Zeit zwischen Ausfällen – der Durchschnittswert zwischen den Ausfällen auf einem Server. Wird mithilfe der folgenden Faktoren berechnet (wobei davon ausgegangen wird, dass der Produktionsserver rund um die Uhr in Betrieb ist):

Laufzeit – die gesamte Zeitspanne, die in den Windows-Ereignisprotokollen abgedeckt wird, während auf dem Server eine bestimmte Betriebssystemversion ausgeführt wird. Da zum Einschätzen der Verfügbarkeit lange Laufzeiten erforderlich sind, wird die Laufzeit häufig in der Maßeinheit "Jahre" ausgedrückt.

Verfügbarkeit – die Wahrscheinlichkeit, dass ein Server die entsprechend vorgesehene Funktion *gegebenenfalls* unter normalen Betriebsbedingungen ausführt. Die Angabe erfolgt in der Regel in Prozent. Die Verfügbarkeit wird mithilfe der folgenden Faktoren berechnet, wobei davon ausgegangen wird, dass der Produktionsserver rund um die Uhr in Betrieb ist:

Durchschnittliche Zeit zum Wiederherstellen – der arithmetische Mittelwert der Ausfalldauer. Als Maßeinheit für die durchschnittliche Zeit zum Wiederherstellen werden Minuten verwendet.

Serververfügbarkeit in Beziehung zur Uhrzeit

Im Allgemeinen dienen Websites der globalen Zielgruppe des Internets. Dies bedeutet, dass die Server, die als Hosts dieser Websites dienen, rund um die Uhr verfügbar sein müssen. Vor diesem Hintergrund kann der metrische Wert für die Verfügbarkeit verwendet werden, um zu berechnen, wie viele Ausfallzeiten ein Server unterstützen und dennoch das geforderte Verfügbarkeitsziel erreichen kann.

In Tabelle 1 wird die Beziehung zwischen der Verfügbarkeit eines Einzelservers und der Ausfallzeit für häufig verwendete Verfügbarkeitsstufen dargestellt. In der Spalte für die Ausfallzeit werden die in einem Jahr maximal zulässigen Ausfallzeiten aufgeführt, mit denen trotzdem das Serververfügbarkeitsziel erreicht werden kann. In den letzten beiden Spalten wird jeweils die in einem Jahr maximal zulässige Anzahl von Ausfällen angezeigt, die bei einer durchschnittlichen Ausfallzeit von 5 und 10 Minuten weiterhin das Erreichen des Serververfügbarkeitszieles ermöglicht. Wenn z. B. eine Verfügbarkeit von 99,9 % auf einem Server erreicht werden soll, darf der Server nicht länger als 8,76 Stunden pro Jahr ausfallen. Im Hinblick auf die Ausfälle entspricht dies maximal 50 bzw. 25 Ausfällen pro Jahr, je nachdem, ob die Wiederherstellungszeit mit 5 bzw. 10 Minuten angegeben ist.

Tabelle 1: Gegenüberstellung von Verfügbarkeit und Ausfallzeit

Verfügbarkeit eines Einzelservers (in Prozent)	Ausfallzeit pro Jahr	Maximal zulässige Ausfälle pro Jahr (bei einer Wiederherstellungszeit von 5 Minuten)	Maximal zulässige Ausfälle pro Jahr (bei einer Wiederherstellungszeit von 10 Minuten)
99,9 %	8,76 Stunden	50	25
99,95 %	4,38 Stunden	25	12
99,99 %	52 Minuten	10	5
99,999 %	5 Minuten	1	0,5*

Anmerkung Wenn eine Verfügbarkeit eines Einzelservers von 99,999 % erreicht werden soll, ist bei einer 5-minütigen Wiederherstellungszeit 1 Ausfall pro Jahr zulässig. Ohne das Integrieren einer fehlertoleranten Hardware- und Softwareredundanz in die jeweilige Umgebung wird dieses Ziel nur schwer zu erreichen sein.

Möglicherweise ist Ihnen im vorherigen Abschnitt aufgefallen, dass es zwei Hauptvariablen für die Berechnung der Verfügbarkeit gibt: die *durchschnittliche Zeit zwischen Ausfällen* und die *durchschnittliche Zeit zum Wiederherstellen*. Wenn Sie die Beziehung zwischen diesen beiden Variablen verstanden haben, können Sie das Verfügbarkeitsziel einfacher erreichen.

Im Allgemeinen kann einer der zwei folgenden Typen von Serverausfällen auftreten: geplante oder ungeplante Ausfälle. Geplante Ausfälle werden von den IT-Abteilungen für die routinemäßige, proaktive Serverwartung budgetiert. Dies kann das Installieren des aktuellen Windows Service Packs oder das Installieren einer neuen Festplatte umfassen, um den Speicheranforderungen eines Servers zu entsprechen. Ungeplante Ausfälle werden nicht im Voraus von IT-Abteilungen budgetiert. Beispiele hierfür umfassen Serverabstürze, Notfallhardwareinstallationen zum Ersetzen einer ausgefallenen Komponente, nicht reagierende Server oder Notfallinstallationen eines Anwendungshotfixes zum Lösen eines kritischen Problems.

Viele Organisationen entscheiden sich für monatliche wartungsbedingte Ausfälle auf den Servern, um notwendige Hardware- oder Softwareupgrades vorzunehmen. Wenn Ihre Organisation Ausfälle plant, ist es von großer Bedeutung, dass Sie wissen, wie sich der monatliche Ausfall auf die Fähigkeit, das gesetzte Verfügbarkeitsziel zu erreichen, auswirkt. In Tabelle 2 wird die zum Erreichen des Verfügbarkeitszieles pro Ausfall *maximal zulässige durchschnittliche Zeit zum Wiederherstellen* aufgeführt. Bei den Berechnungen wird eine durchschnittliche Zeit von 60 Tagen zwischen den Ausfällen verwendet.

Tabelle 2: Maximal zulässige durchschnittliche Zeit zum Wiederherstellen

Verfügbarkeit eines Einzelservers (in Prozent)	Durchschnittliche Zeit zwischen Ausfällen (in Tagen)	Maximal zulässige durchschnittliche Zeit zum Wiederherstellen (in Minuten)
99,9 %	60	86,5
99,95 %	60	43,2
99,99 %	60	8,6
99,999 %*	60	0,9

Anmerkung Bei einer durchschnittlichen Zeit von 60 Tagen zwischen den Ausfällen ist es unmöglich, ein Serververfügbarkeitsziel von 99,999 % zu erreichen, weil die Serverstartzeit den Höchstwert von 0,9 Minuten für die *durchschnittliche Zeit zum Wiederherstellen* übersteigen würde.

Bei den in dieser Studie überwachten Websites mit hohen Anforderungen lag die *durchschnittliche Zeit zum Wiederherstellen* für Webserver zwischen 5 und 10 Minuten. Diese durchschnittliche Zeit zum Wiederherstellen kann erreicht werden, wenn eine Organisation in der Lage ist, Fehler schnell zu beheben und die Webserver schnell wieder in Betrieb zu setzen.

In Tabelle 3 wird die zum Erreichen des Verfügbarkeitszieles minimal zulässige durchschnittliche Zeit zwischen Ausfällen angegeben, wobei die durchschnittliche Zeit zum Wiederherstellen 10 Minuten beträgt. Die Berechnungen in Tabelle 3 umfassen sowohl geplante als auch ungeplante Ausfälle. Wenn sich Ihre Organisation entscheidet, die während einer Wartung auftretenden Ausfälle nicht zu berücksichtigen, müssen Sie die Verfügbarkeitsfaktoren neu berechnen, wobei Sie die für die Wartung geplante Ausfallzeit entfernen.

Tabelle 3: Minimal zulässige durchschnittliche Zeit zwischen Ausfällen

Verfügbarkeit eines Einzelservers (in Prozent)	**Durchschnittliche Zeit zwischen Ausfällen (in Tagen)**	**Durchschnittliche Zeit zum Wiederherstellen (in Minuten)**
99,9 %	7	10
99,95 %	14	10
99,99 %	70	10
99,999 %	694	10

Aus Tabelle 3 geht hervor, dass die Server mehr als 70 Tage lang fehlerfrei ausgeführt werden müssen, um ein Serververfügbarkeitsziel von über 99,99 % zu erreichen, wobei von einer durchschnittlichen Zeit zum Wiederherstellen von 10 Minuten ausgegangen wird. Ohne die Verwendung redundanter Hardware ist das Erreichen einer Verfügbarkeit eines Einzelservers von 99,999 % nicht möglich.

Abschließend muss festgehalten werden, dass das Erreichen des gewünschten Serververfügbarkeitszieles seinen Preis hat. Die Tabellen 2 und 3 wurden einbezogen, um die Beziehung zwischen den zwei Elementen der Serververfügbarkeit – die durchschnittliche Zeit zwischen Ausfällen und die durchschnittliche Zeit zum Wiederherstellen pro Ausfall – aufzuzeigen. Bei unseren Messungen haben wir zwar herausgefunden, dass die durchschnittliche Zeit zum Wiederherstellen 5 bis 10 Minuten beträgt, dies soll jedoch nicht implizieren, dass alle Ausfälle 10 Minuten dauern. Die Aufteilungen der gemessenen Serverausfälle enthalten auch immer einen kleinen Anteil sehr langer Ausfälle, obwohl die meisten Ausfälle unter 10 Minuten liegen. Beim Untersuchen der Verhältnisse zwischen den Komponenten der Serververfügbarkeit werden Sie erkennen, dass Ihr IT-Unternehmen die Serververfügbarkeit durch eine bestimmte Zeitplanung beeinflussen kann, um die entsprechenden Verfügbarkeitsziele zu erreichen.

Wenn die Serververfügbarkeitsziele von den angegebenen Beispielen abweichen, können Sie die unten aufgeführten Formeln verwenden, um die erforderlichen Werte für die durchschnittliche Zeit zwischen Ausfällen und die durchschnittliche Zeit zum Wiederherstellen zu ermitteln, damit das Verfügbarkeitsziel erreicht werden kann. Im angegebenen Beispiel wird ein Serververfügbarkeitsziel von 99,97 % vorausgesetzt, wobei die durchschnittliche Zeit zwischen den Ausfällen im ersten Beispiel 60 Tage und die durchschnittliche Zeit zum Wiederherstellen im zweiten Beispiel 10 Minuten beträgt. Diese Beispiele sind eingeschlossen, damit Sie die mathematischen Formeln verstehen.

Ermitteln der durchschnittlichen Zeit zum Wiederherstellen, wenn die durchschnittliche Zeit zwischen Ausfällen angegeben ist:

Durchschnittliche Zeit zum Wiederherstellen (Min.) = Durchschnittliche Zeit zwischen Ausfällen (Min.) * (1-Verfügbarkeit)/Verfügbarkeit
Durchschnittliche Zeit zum Wiederherstellen (Min.) = 86400 Minuten * (1-0,9997)/0,9997))
Durchschnittliche Zeit zum Wiederherstellen (Min.) = 86400 Minuten * (0,0003/0,9997)
Durchschnittliche Zeit zum Wiederherstellen (Min.) = 25,93 Minuten

Ermitteln der durchschnittlichen Zeit zwischen Ausfällen, wenn die durchschnittliche Zeit zum Wiederherstellen angegeben ist:

> Durchschnittliche Zeit zwischen Ausfällen (Min.) = Durchschnittliche Zeit zum Wiederherstellen (Min.) * (Verfügbarkeit/(1-Verfügbarkeit))
> Durchschnittliche Zeit zwischen Ausfällen (Min.) = 10 Min. * (0,9997/(1-0,9997))
> Durchschnittliche Zeit zwischen Ausfällen (Min.) = 33323 Minuten
> Durchschnittliche Zeit zwischen Ausfällen (Tage) = 33323 Minuten / 24 Stunden / 60 Minuten
> Durchschnittliche Zeit zwischen Ausfällen (Tage) = 23,1 Tage

Tools zum Messen der Verfügbarkeit

Die Microsoft Reliability Group hat ein Tool namens Event Log Analyst (ELA) entwickelt, das Informationen zur Zuverlässigkeit von Remoteservern unter Windows 2000 und Windows NT 4.0-basierten Servern sammelt. ELA wird auf einem einzelnen Computer (dem so genannten "Sammlungscomputer") ausgeführt und ruft fortlaufend Ereignisprotokollinformationen von anderen Windows 2000- und NT 4.0-basierten Servern ab. Nachdem die ELA-Daten gesammelt wurden, können sie mithilfe anderer Tools analysiert werden.

ELA sammelt unterschiedliche Arten grundlegender Informationen aus den System- und Anwendungsereignisprotokollen. Die gesammelten Daten umfassen Folgendes:

- Grundlegende Informationen zu einem Windows 2000-Ereignisprotokoll: die Anzahl von Systemereignissen und die Anzahl von Anwendungsereignissen.
- Zeitstempel aller Systemneustarts. Auf Computern unter Windows NT 4.0 Service Pack 4 oder höher werden außerdem die Zeitstempel der Ereignisse beim Herunterfahren des Systems gesammelt.
- Informationen zu Windows 2000-Systemabstürzen (werden auch als Fehlerüberprüfungen oder Bluescreens bezeichnet).
- Informationen zu Anwendungsabstürzen (werden auch als Speicherabbilder im Benutzermodus oder Dr. Watson-Benachrichtigungen bezeichnet).

ELA ist auf eine unauffällige Struktur ausgelegt. Da ELA nur öffentlich dokumentierte Schnittstellen für den Remotezugriff auf das Ereignisprotokoll verwendet, muss auf den Systemen, auf denen sich die Ereignisprotokolle befinden, keine Software installiert werden. ELA kann ganz problemlos installiert werden. Es handelt sich um eine einzelne ausführbare Datei, die auf jedem System unter Windows 2000 oder höher ausgeführt werden kann. ELA wirkt sich nur geringfügig auf Produktionsumgebungen aus, wobei in der Regel während des Sammlungsprozesses weniger als 30 Sekunden auf den Remoteserver zugegriffen wird. In Tests mit dem unternehmenseigenen Rechenzentrum von Microsoft hat ELA die Ereignisprotokolle von über 1.500 Servern in ca. 30 Minuten mithilfe eines Sammlungssystems mit einer LAN-Verbindung von 100 Mbit/s durchsucht.

Das umfassende ELA-Benutzerhandbuch beeinhaltet eine Liste mit gesammelten Standardereignissen sowie eine Sicherheitskonfiguration für das Sammeln von Daten (siehe Anhang).

Empfohlene Vorgehensweisen für das Entwerfen von Webanwendungen

Beim Erstellen einer Webanwendung mit hoher Verfügbarkeit müssen viele Auswahlmöglichkeiten für den Entwurf berücksichtigt werden. IIS stellt unendlich viele Möglichkeiten für einen Entwickler in Form von Tools, Sprachen, Datenbankverbindungen und Skriptoptionen bereit. Es ist wichtig, dass die Entwickler die mit jeder einzelnen Option verbundenen Vorteile und Nachteile kennen. Im nächsten Abschnitt wird die Entwicklung empfohlener Vorgehensweisen erläutert, damit Sie die Qualität der vorhandenen Anwendungen einfacher verbessern können und Möglichkeiten kennen lernen, um künftig bessere Webanwendungen erstellen zu können.

Komponenten

In den in diesem Abschnitt aufgeführten Artikeln werden empfohlene Vorgehensweisen für das Entwerfen und Konfigurieren von Komponenten umrissen. Beispiele für Komponenten umfassen eine MTS-Komponente (Microsoft Transaction Server), eine COM+-Komponente oder eine von Active Server Pages (ASP) aufgerufene Komponente. Beim Schreiben von Komponenten mit Visual Basic (VB) müssen Sie bestimmte Eigenschaften im VB-Projekt festlegen. Außerdem sollten Sie die Empfehlungen für optimale Vorgehensweisen befolgen, um die Leistungsfähigkeit zu verbessern.

Entwickler deklarieren z. B. häufig öffentliche Variablen in Basismodulen (BAS) mithilfe von Microsoft Visual Basic. Diese Vorgehensweise kann zu einem unvorhersehbaren Verhalten und zu potenziellen Speicherbeschädigungen führen, wodurch die Wahrscheinlichkeit von Anwendungsabstürzen erhöht wird. Verwenden Sie die in diesem Abschnitt genannten Artikel, um ein besseres Verständnis der Maßnahmen zu erlangen, die Sie beim Entwerfen und Konfigurieren von Komponenten ergreifen können.

Vermeiden Sie das Verwenden öffentlicher Variablen in Basismodulen (BAS). Dies kann ebenfalls zu unvorhersehbaren Verhalten und Speicherbeschädigungen führen, da jeder Thread über eine separate Kopie der Variablen verfügt, während die Anforderungen auf diesem Thread dieselben Variablen verwenden.

Stellen Sie sicher, dass die Kontrollkästchen **In Speicher erhalten** und **Unbeaufsichtigte Ausführung** aktiviert sind, wenn Sie eine serverseitige Komponente mithilfe von Visual Basic schreiben. Wenn diese Optionen nicht aktiviert sind, entlädt das Visual Basic-Laufzeitmodul unerwartet benutzerdefinierte und Laufzeit-DLLs, so dass der Computer möglicherweise nicht mehr in Multithreadszenarien reagiert (es kommt also zum Absturz oder Stillstand).

Verwenden Sie möglichst häufig gepoolte ASP-Seiten und Komponenten. COM-Komponenten verfügen über drei Isolierungsoptionen: nicht konfiguriert, als Bibliotheksanwendung konfiguriert und als Serveranwendung konfiguriert. Nicht konfigurierte Anwendungen und Bibliotheksanwendungen werden im Prozessbereich der Aufruffunktion oder "prozessintern" ausgeführt, während Serveranwendungen für die Ausführung im eigenen Prozessbereich konfiguriert sind. Mithilfe der Prozessisolation in ASP und COM wird vermieden, dass ein Absturz einer Anwendung weitere Abstürze der anderen installierten Anwendungen verursacht.

In den unten aufgelisteten Artikeln werden empfohlene Vorgehensweisen für das Erstellen von Komponenten beschrieben:

- ASP-Komponentenrichtlinien bzw. ASP Component Guidelines (englischsprachig)
- Q281630 HOWTO Configure Visual Basic DLL to Run in IIS, MTS, or COM+ (englischsprachig)
- Design Guidelines for VB Components Under ASP (englischsprachig)
- Developing a Visual Basic Component for IIS/MTS (englischsprachig)

ASP – Empfohlene Vorgehensweisen und häufig auftretende Probleme

ASP-Abstürze zählen zu den 10 häufigsten Problemszenarien, auf die Entwicklersupportgruppen zu Microsoft IIS stoßen. Obwohl es zahlreiche Gründe dafür geben kann, ist in vielen Fällen der Kundenanwendungscode die eigentliche Ursache des Problems oder zumindest ein wichtiger Faktor. Diese Art von Problemen kann beim Versuch, den Code auf einem Produktionsserver zu debuggen, zu längeren Ausfallzeiten führen. Bei einem einfachen Beispiel von Code, der einen Absturz verursachen kann, erfolgen explizite Aufrufe der Methoden **Application.Lock** und **Application.Unlock** innerhalb einer Komponente, die von einer ASP-Seite aus aufgerufen wird. Dies lässt sich folgendermaßen veranschaulichen:

```
Application.Lock
Application ("myvar") = "Hello World"
Application.Unlock
```

Im obigen Beispiel sind die Methoden **Application.Lock** und **Application.Unlock** nicht zum Zuweisen einer einzelnen Anwendungsvariable erforderlich, weil diese Methoden implizit aufgerufen werden.

Konfigurieren Sie Ihre ASP-Anwendungen für die Verwendung der Prozessisolation, um zu vermeiden, dass sich Abstürze einer Anwendung auf andere installierte Anwendungen auswirken. Die drei in IIS 5.0 verfügbaren Isolationsstufen sind **Niedrig**, **Mittel** und **Hoch**. Bei der niedrigen Isolation, die häufig auch als "prozessinterne" Isolation bezeichnet wird, werden die ASP-Anwendungen innerhalb des Prozesskontextes von **inetinfo.exe** ausgeführt. Hierbei handelt es sich um den primären IIS-Prozess. Die mittlere Isolation wird als "prozessextern" bezeichnet, weil ASP außerhalb des IIS-Prozesses ausgeführt wird. Alle für die Ausführung auf mittlerer Isolationsstufe konfigurierten ASP-Anwendungen verwenden einen gemeinsamen Prozessbereich. Bei der hohen Isolation, die ebenfalls als "prozessextern" bezeichnet wird, werden alle ASP-Anwendungen in ihrem eigenen Prozessbereich ausgeführt, wodurch die ASP-Anwendungen jeweils voreinander geschützt werden.

Verwenden Sie das Web Capacity Analysis-Tool und das Tool für die HTTP-Überwachung aus dem IIS Resource Kit, um Belastungstests und Messungen der Webanwendungen durchzuführen. Das Durchführen von Belastungstests der Webanwendungen ist nicht nur beim Isolieren von Codierungsfehlern hilfreich, sondern es ermöglicht Ihnen außerdem das Ermitteln möglicher Engpässe, die Sie gegebenenfalls optimieren müssen.

In den folgenden Artikeln werden empfohlene Vorgehensweisen für das Schreiben von ASP-Anwendungen beschrieben:

- Über 25 ASP-Tipps zum Verbessern von Leistung und Stil bzw.
  25+ ASP Tips to Improve Performance and Style (englischsprachig)
- Maximizing ASP Performance (englischsprachig)

Empfohlene Vorgehensweisen für ASP-Datenbanken

Das Abrufen von Daten aus Microsoft SQL Server mithilfe von ADO (ActiveX Data Objects) und ASP kann für Webentwickler eine Herausforderung darstellen. Da eine zunehmende Anzahl von Webanwendungen als Schnittstelle für Datenbanken dient, ist es von Bedeutung, dass Sie Wege kennen, um bei Ihren Entwicklungsbemühungen die Leistungsfähigkeit, Skalierbarkeit und Robustheit zu maximieren. Verwenden Sie die folgenden Richtlinien und empfohlenen Vorgehensweisen, wenn Sie das Aufrufen von Datenbankkomponenten aus Webanwendungen in IIS planen:

Schließen Sie bei ADO grundsätzlich sämtliche Recordsets und Verbindungen.

Wenn Sie eine Verbindung "öffnen", verwenden Sie die Verbindung, und "schließen" Sie sie anschließend. Die Verbindung kann dann sicher an einen anderen Thread weitergeleitet werden, der einen anderen Befehl verarbeitet. Wenn sich die Serverauslastung verringert, wird der Verbindungspool automatisch verkleinert, und andere Serverbenutzer erhalten eine bessere Leistung. Wenn die Serverauslastung stark zunimmt, kann der Pool gegebenenfalls vergrößert werden. Sofern Verbindungen nicht zusammengelegt werden, führt dies zu Leerlaufverbindungen, die wiederum unnötig Server- und Netzwerkressourcen verbrauchen. Darüber hinaus stoßen Sie möglicherweise auch auf Threadingprobleme, die auftreten können, wenn mehrere gleichzeitige Threads dieselbe Verbindung verwenden.

- Q176056: ADO/ASP Scalability FAQ (englischsprachig)

Spätes Öffnen, frühzeitiges Schließen.

Öffnen Sie ADO-Objekte erst kurz bevor sie benötigt werden, und schließen Sie die Objekte unmittelbar nach Beendigung des jeweiligen Vorgangs. Dadurch wird die Zeitspanne verkürzt, in der die Datenbank mit den Ressourcen "jonglieren" muss, und die Datenbankverbindung wird so schnell wie möglich für den Verbindungspool freigegeben, um neue Verbindungen zuzulassen.

- Über 25 ASP-Tipps zum Verbessern von Leistung und Stil bzw.
  25+ ASP Tips to Improve Performance and Style (englischsprachig)

Übergeben Sie keine Parameter an das Befehlsobjekt in der EXECUTE-Anweisung. Durch das Übergeben von Parametern an das Befehlsobjekt wird ADO gezwungen, zusätzliche Verarbeitungsaufgaben zu bewältigen und Rückschlüsse auf die von Ihnen übergebenen Parameter zu ziehen.

Im folgenden Codebeispiel wird die ungünstige Übergabe von Parametern an das Befehlsobjekt veranschaulicht.

```
Set DB = Server.CreateObject ("ADODB.Connection")
DB.Open "Provider=SQLOLEDB;Data
Source=dsnTest;Database=dbTest;UserId=john;Password=doe;"
Set RS = DB.Execute ("GetCustomerByLastName @LastName='Smith'")
RS.Close
DB.Close
Set RS = Nothing
Set DB = Nothing
```

Eine bessere Vorgehensweise besteht darin, die Parameter für das Befehlsobjekt explizit zu deklarieren, wie im Codebeispiel angezeigt.

```
Set DB = Server.CreateObject ("ADODB.Connection")
DB.Open "Provider=SQLOLEDB;Data
Source=dsnTest;Database=dbTest;UserId=john;Password=doe;"
Set cmdTemp = Server.CreateObject ("ADODB.Command")
cmdTemp.ActiveConnection = DB
cmdTemp.CommandText = "GetCustomerByLastName"
cmdTemp.CommandType = adCmdStoredProc
Set params = cmdTemp.Parameters
params.Append cmdTemp.CreateParameter ("RETURN_VALUE", adInteger,
adParamReturnValue, 0)
params.Append cmdTemp.CreateParameter ("@LastName", adVarChar,
adParamInput,
15)
cmdTemp ("@LastName") = "Smith"
Set RS = cmdTemp.Execute
RS.Close
DB.Close
Set RS = Nothing
Set DB = Nothing
Set cmdTemp = Nothing
```

*Verwenden Sie grundsätzlich **Server.CreateObject**.*

Durch das Verwenden von **Server.CreateObject** kann ASP die Objektinstanz nachverfolgen. Der serverseitige Prozess sorgt dafür, dass das Objekt in einem Transaction Server-Paket erstellt wird, so dass die Ressourcen in einem Pool zusammengefasst werden. In serverseitigen Skripts ermöglicht das Verwenden der Funktionen **CreateObject** und **GetObject** anstelle von **Server.CreateObject** weder den Zugriff auf integrierte ASP-Objekte noch die Teilnahme an Transaktionen. Beim Verwenden von **CreateObject** und **GetObject** wird jedes neue Objekt an einen separaten Thread angefügt. Dabei werden die verfügbaren Systemressourcen viel schneller verbraucht als unter Verwendung der verfügbaren Verbindungspoolingfunktionen mithilfe von **Server.CreateObject**.

- ASP-Komponentenrichtlinien bzw. ASP Component Guidelines (englischsprachig)

Verwenden Sie Recordset- oder Befehlsvariablen nicht mehrmals, sondern erstellen Sie jeweils neue.

Das wiederholte Verwenden von Recordset- oder Befehlsvariablen kann das Risiko erhöhen, dass der Code einen Fehler innerhalb von ADO verursacht. Das Befehlsobjekt ist für diese Art von Verwendung nicht vorgesehen.

- Q197449: Problems Reusing ADO Command Object on Multiple Recordsets (englischsprachig)

Verwenden Sie beim Konfigurieren von ODBC-Einstellungen für die entsprechende Datenquelle möglichst viele System-DSNs (anstelle von Datei-DSNs). Ein System-DSN ist dreimal schneller als ein Datei-DSN.

Verwenden Sie ADO-Verbindungen nicht in Sitzungsobjekten.

Wenn ADO-Objekte in Sitzungen eingesetzt werden, kommt es zu Skalierbarkeitseinschränkungen und Threadingproblemen sowie zu einer unnötig hohen Belastung auf dem Webserver und in der Datenbank. Wenn eine Verbindung in einer Sitzungsvariablen gespeichert wird, entfällt das Verbindungspooling, weil die im Sitzungsobjekt gespeicherten Variablen die gesamte Benutzersitzung über weiter bestehen. Das Verbindungspooling zahlt sich aus, wenn die Verbindungen auf mehreren Clients gemeinsam genutzt und die Ressourcen nur so lange verwendet werden wie erforderlich. Ein in einer Sitzungsvariablen gespeichertes Verbindungsobjekt dient nur dem Benutzer, für den die Sitzung erstellt wurde, und die Verbindung wird erst bei Beendigung der Sitzung für den Pool freigegeben.

- Top Ten Tips: Accessing SQL Through ADO and ASP (englischsprachig)
- MSDN Library: Session Object (englischsprachig)

Verwenden Sie TCP/IP-Sockets zum Herstellen einer Verbindung mit SQL Server, wenn das Programm auf einem Remotecomputer ausgeführt wird.

TCP/IP-Sockets erfordern keine vertrauenswürdige NT-Verbindung und verwenden die SQL-Standardsicherheit, wobei die mit der Verwendung von Named Pipes für einen Remotecomputer verbundenen Authentifizierungsprobleme umgangen werden (siehe unten). In Fällen, in denen sich SQL Server auf einem anderen Computer befindet, bieten TCP/IP-Sockets eine schnellere Verbindung.

- Top Ten Tips:Accessing SQL Through ADO and ASP (englischsprachig)

Verwenden Sie Named Pipes, wenn SQL Server lokal (auf demselben Computer wie ASP) ausgeführt wird.

Standardmäßig ist für die Verwendung von Named Pipes im Netzwerk eine vertrauenswürdige Verbindung erforderlich. Wenn ein Client versucht, eine Verbindung über Named Pipes im Netzwerk herzustellen, führt der Computer unter Windows NT mit SQL Server eine Sicherheitsprüfung durch und muss das Clientcomputerkonto authentifizieren, was wiederum eine Rundsendung zum entsprechenden Domänencontroller erfordert. Wenn der Pfad zwischen SQL Server und dem Domänencontroller nicht verfügbar ist, kann möglicherweise keine Verbindung hergestellt werden.

Sofern sich SQL Server auf demselben Computer befindet, auf dem auch IIS ausgeführt und ASP verwaltet werden, sollten Sie eine lokale Named Pipe-Verbindung anstelle einer Named Pipe-Netzwerkverbindung verwenden. Dies kann problemlos ausgeführt werden, indem Sie einfach in der Datei **global.asa** in der SQL Server-Verbindungszeichenfolge das Stichwort SERVER=machinename in SERVER=<lokal> ändern. Dadurch wird die Rundsendung zum Domänencontroller zwecks Authentifizierung vermieden, und die wertvolle Netzwerkbandbreite kann eingespart werden.

- Q207671: How to: Accessing Network Files from IIS Application (englischsprachig)

Stellen Sie sicher, dass die Verbindungsobjekte in jeder einzelnen ASP-Seite erstellt werden.

Beim Erstellen und Löschen des Verbindungsobjekts pro Seite (im Gegensatz zum Speichern des Objekts in einer Sitzungsvariablen) werden die Vorteile des Verbindungspoolings genutzt. Durch Löschen des Objekts auf jeder einzelnen Seite wird die Verbindung für den Pool freigegeben, so dass sie erneut verwendet werden kann. Dank dieser gemeinsamen Ressourcennutzung wird die Auslastung des Servers reduziert, und auch die Zeit zum Herstellen einer Datenbankverbindung kann für die Benutzer verkürzt werden, nachdem die ersten Verbindungen erstellt wurden. In einem komplexen ISAPI-Szenario hat die Microsoft Performance Group mithilfe einer optimierten Implementierung von Verbindungspooling eine Verbesserung der CPU-Verwendung von 30 bis 40 % festgestellt. Diese Messung erfolgte mithilfe von Leistungsmaßstäben, wie TPC-C und TPC-W.

- Top Ten Tips:Accessing SQL Through ADO and ASP (englischsprachig)

Weitere Hyperlinks und Informationen zu Ressourcen, die sich auf die Webanwendungsentwicklung beziehen, finden Sie im Anhang.

Webserver-Bereitstellungsplanung

Nachdem Sie die Webanwendung vollständig entworfen und entwickelt haben, ist es an der Zeit, die Webserver zu erstellen und zu testen, auf denen die neue Anwendung verwaltet wird. Neben den erforderlichen Schritten zum Installieren von Windows 2000 müssen Sie außerdem sicherstellen, dass alle von Ihnen verwendeten Supportanwendungen sowie die physische Hardware mit Windows 2000 kompatibel sind. In den nächsten Abschnitten werden wichtige Aufgaben zur Planung beschrieben, die durchgeführt werden müssen, um eine Webserverplattform mit hoher Verfügbarkeit sicherstellen zu können.

Anwendungskompatibilität

Einer der ersten Schritte, den Sie beim Installieren von oder Aktualisieren auf Windows 2000 durchführen sollten, besteht darin, sicherzustellen, dass die von Ihnen zum Unterstützen des Unternehmens verwendeten Anwendungen mit Windows 2000 kompatibel sind. Durch das Zusammenarbeiten mit externen Kunden hat Microsoft herausgefunden, dass die mit Windows 2000 inkompatiblen Anwendungen eine eine Hauptursache darstellen für Probleme, die zu ungewünschten Ausfallzeiten und einer verminderten Serververfügbarkeit führen.

Es gibt ein breites Spektrum von Symptomen, die mit inkompatiblen Anwendungen verbunden sind, wie z. B. die Unfähigkeit der Anwendung, zu starten, Arbeitsspeicherverluste und Zugriffsverletzungen.

Microsoft stellt ein Onlinetool namens Windows 2000 Readiness Analyzer bereit, das zum Erstellen eines Berichts verwendet werden kann, in dem bekannte Hardware- und Softwarekompatibilitätsprobleme der bestimmten Server detailliert aufgeführt werden. An dieser Stelle sollte noch darauf hingewiesen werden, dass die Kompatibilitätsprobleme bei den Produkten von Drittanbietern nicht mit diesem Tool erfasst werden. Dies bezieht sich vor allem auf Anwendungen, die Sie selbst entwickelt und bereitgestellt haben.

Empfohlene Vorgehensweise – Identifizieren und lösen Sie Anwendungskompatibilitätsprobleme vor dem Aktualisieren auf Windows 2000 mithilfe der in den unten aufgeführten Artikeln enthaltenen Informationen. Der Assistent zum Suchen kompatibler Softwareanwendungen und Microsoft Readiness Analyzer ermöglichen Ihnen das Suchen nach kompatiblen Anwendungen für unterschiedliche Versionen von Windows 2000. Das Anwendungskompatibilitäts-Toolkit enthält Whitepaper und Tools, um Kunden von Microsoft Windows beim Diagnostizieren und Lösen von Anwendungskompatibilitätsproblemen zu unterstützen. Dies umfasst ein Whitepaper mit häufig auftretenden Kompatibilitätsproblemen sowie mehrere Dokumente über empfohlene Vorgehensweisen bei Tests und Tools zum Beheben von Kompatibilitätsproblemen.

- Search for Compatible Software Applications (englischsprachig)
- Microsoft Windows 2000 Readiness Analyzer (englischsprachig)
- Windows Application Compatibility Toolkit (englischsprachig)

Hardwarekompatibilität

Beim Aktualisieren auf oder Installieren von Windows 2000 ist auch das Berücksichtigen der Hardwarekompatibilität sehr wichtig. Microsoft hat häufig über Kernursachenanalysen herausgefunden, dass viele Fehlerereignisse, wie z. B. Bluescreens, aufgrund fehlerhafter Filtertreiber oder eines inkompatiblen BIOS auftreten. Zahlreiche Dotcom-Unternehmen, die mit dem Microsoft Reliability Team kooperieren, haben berichtet, dass in bestimmten Situationen beim Installieren von Windows 2000 ungeplante Ausfälle (Bluescreens) aufgetreten sind, weil der Treiber für die SCSI-Geräte nicht auf den Windows 2000-Medien vorhanden war. Glücklicherweise stellt Microsoft jedoch viele Ressourcen bereit, so dass Sie die Inkompatibilität von Hardware bereits identifizieren und beheben können, bevor Sie die Webserver in Ihrer Produktionsumgebung installieren.

Das im vorherigen Abschnitt erwähnte Tool Windows 2000 Readiness Analyzer kann Sie auch dabei unterstützen, potenzielle Hardwarekompatibilitätsprobleme auf den Servern zu ermitteln. Wenn Sie von Windows NT 4.0 auf Windows 2000 aktualisieren, können Sie alternativ dazu auch Kompatibilitätsprobleme bei der Serverkonfiguration identifizieren, indem Sie nach dem Einlegen der CD-Rom für das Windows 2000-Setup in der Befehlszeile den Befehl **winnt32 /checkupgradeonly** ausführen.

Neben der Nutzung der von Microsoft bereitgestellten Tools und Ressourcen ist es außerdem wichtig, dass Sie sich direkt mit den Serveranbietern in Verbindung setzen, um die Produktkompatibilität mit Windows 2000 zu ermitteln. Hardwarehersteller geben oft BIOS-Aktualisierungen und Treiberaktualisierungen frei, die als Windows 2000-kompatibel getestet und zertifiziert wurden. Es ist von großer Bedeutung, dass Sie sämtliche Treiberaktualisierungen der Hardwarehersteller vor dem Installieren von Windows 2000 downloaden, testen und installieren.

Empfohlene Vorgehensweise – Stellen Sie die Hardware- und Treiberkompatibilität mit Windows 2000 vor der Installation mithilfe der im Folgenden aufgeführten Links sicher. Mithilfe der Assistenten zum Suchen kompatibler Hardwarekomponenten und Computer können Sie vor dem Aktualisieren oder Installieren ermitteln, ob Ihre Server und die verwendete Hardware mit Windows 2000 kompatibel sind. Einige Suchergebnisse umfassen auch Hyperlinks zum Downloaden von Windows 2000-Treibern, die von einigen Herstellern bereitgestellt werden.

- Search for Compatible Hardware Devices (englischsprachig)
- Search for Compatible Computers (englischsprachig)

Überlegungen zur Bereitstellung von Windows 2000

Sobald Sie sicher sind, dass die Serverhardware und die Anwendungen mit Windows 2000 kompatibel sind, können Sie Windows 2000 auf den Webservern bereitstellen. Sie sollten mehrere Optionen berücksichtigen, wenn Sie sich für die für Ihre jeweilige Geschäftsumgebung am besten geeignete Bereitstellungsmethode entscheiden. In den nächsten Abschnitten wird der Servererstellungsprozess umrissen, den Sie befolgen sollten, um eine Plattform mit hoher Verfügbarkeit sicherstellen zu können.

Im Allgemeinen sollte das Erstellen von Servern folgende Schritte umfassen:

1. Ermitteln der Hardwareanforderungen (Datenträgereinrichtung, Arbeitsspeicher usw.)
2. Ermitteln der Anwendungsanforderungen (zusätzliche Software, Skriptumgebungen usw.)
3. Identifizieren der Installationsoptionen (abbildbasiert, Skriptinstallation, RIS in Whistler)
4. Festlegen der erforderlichen Betriebssystemkonfiguration vor der Datenträgerduplizierung (sofern vorgesehen)
5. Dokumentieren einer konsistenten, wiederholbaren Installationsmethode

Diese Schritte werden im Folgenden detaillierter erläutert.

Schritt 1: Ermitteln der Hardwareanforderungen und der Standardplattform

Aufgrund der Zusammenarbeit mit vielen unterschiedlichen Kunden lag es für Microsoft auf der Hand, dass das Auswählen und Einhalten einer Standardplattform die Installation des Betriebssystems, die Anwendungsinstallation sowie die Problembehandlung für Systemadministratoren erheblich vereinfacht. Beispiele für die Art von Entscheidungen, die hinsichtlich einer einheitlichen Hardwareplattform getroffen werden müssen, umfassen Folgendes:

- Wie sollten die Festplatten partitioniert werden?
- Sollten Festplatten- oder CD-ROM-Laufwerke neu zu verschiedenen Laufwerkbuchstaben zugeordnet werden?
- Benötigen Sie jeweils einen oder mehrere Prozessoren in den Webservern?

Empfohlene Vorgehensweise – Verwenden Sie das Dokument "Web Server Capacity Planning" (englischsprachig), um die oben aufgeführten Fragen sowie viele weitere Fragen zur Hardwarekapazitätsplanung besser beantworten zu können. Vom Hardwaregesichtspunkt aus besteht die sinnvollste Kapazitätsplanungsstrategie darin, die Nutzung sorgfältig zu beobachten, bestimmte Muster zu überwachen und die Ressourcen anhand der Überwachungsergebnisse zu erhöhen oder zu verringern.

- Web Server Capacity Planning (englischsprachig)

Schritt 2: Ermitteln der Anwendungsanforderungen

In den meisten Fällen umfasst eine Webanwendung mehrere Teile, für die häufig unterschiedliche Server als Host dienen. Dies erfordert oft die Installation zusätzlicher Software auf den Webservern. So verwenden Webanwendungen beispielsweise häufig Datenbanken, wenn es um E-Commerce-Transaktionen oder auch darum geht, Möglichkeiten zu finden, um den jeweiligen Saldo der einzelnen Bankkonten abzufragen. Zu diesem Zweck müssen Sie bestimmte Konfigurationen wie Datenbankverbindungsmethoden sowie Netzwerkoptimierungen untersuchen, um eine erfolgreiche Integration der Softwarekomponenten sicherzustellen.

Empfohlene Vorgehensweise – Beim Planen von Servererstellungen ist es wichtig, nicht nur die erforderlichen Anwendungen zu beachten, sondern es müssen auch die Installationsmethoden für diese Anwendungen und die gegebenenfalls zum Unterstützen dieser Anwendungen erforderlichen Zusatzkonfigurationen berücksichtigt werden. Wenn Sie beispielsweise das Verwenden einer SQL-Datenbank als Teil der Webanwendung vorsehen, müssen Sie möglicherweise ODBC installieren und konfigurieren, um Ihre Anwendungsanforderungen zu erfüllen. Diese Informationen sind von großer Bedeutung, wenn die Installation automatisiert und Installationsprobleme behandelt werden sollen.

Schritt 3: Identifizieren der Installationsoptionen

Beim Installieren von Servern unter Windows 2000 Server und Windows 2000 Advanced Server stehen drei Hauptoptionen zur Verfügung. Dabei handelt es sich um die folgenden Optionen:

- Unbeaufsichtigte Installation mithilfe einer startfähigen CD und einer Antwortdatei
- Unbeaufsichtigte Installation mithilfe einer Netzwerkfreigabe und einer Antwortdatei
- Installation eines Datenträgerabbildes mit Sysprep

Jede dieser Installationsoptionen hat ihre Vor- und Nachteile, die detailliert in Tabelle 4 aufgeführt werden.

Tabelle 4: Vergleich der Installationstools

Installationstyp	Vorteile	Nachteile
Unbeaufsichtigte CD-basierte Installation	- Eignet sich für Computer, die nicht an Netzwerke angeschlossen sind - Problemloses Anpassen der Installation	- Das Eingreifen durch den Benutzer ist erforderlich, um die Diskette (oder CD) nach dem Beenden von GUI-Modus Setup auszuwerfen - Potenzielles Sicherheitsrisiko für die Firmeninformationen auf der CD
Unbeaufsichtigte netzwerkbasierte Installation	- Installationsdateien werden auf einem zentralen schreibgeschützten Server gesichert - Kein Benutzereingriff während der Installation erforderlich	- Das Erstellen einer einzelnen Startdiskette mit allen zum Herstellen einer Netzwerkverbindung erforderlichen Tools ist problematisch
Installation eines Datenträgerabbildes mit Sysprep	- Schnellste Option für eine automatisierte Installation - Keine Skripterstellung erforderlich, weil das Abbild das Betriebssystem und Anwendungen enthält	- Die Einhaltung einer strikten Versionskontrolle der Betriebssystemdateien ist schwierig, wenn viele CDs vorhanden sind - In den meisten Fällen ist ein Benutzereingriff zur Eingabe der Product ID erforderlich

Unbeaufsichtigte Installation mit CD

Wie bereits zuvor erwähnt, ist diese Installationsmethode am besten für Szenarien mit geringer Netzwerkbandbreite geeignet (wie z. B. für Zweigstellen an unterschiedlichen Standorten).

Das Verwenden von Antwortdateien ist erforderlich, um Windows 2000 Ihren Anforderungen entsprechend installieren zu können. Bei einer Antwortdatei handelt es sich im Wesentlichen um ein Skript, das die Fragen während der Installation beantwortet, ohne dass eine Benutzereingabe erforderlich ist. Auf der Windows 2000-CD ist eine Vorlagenantwortdatei mit dem Namen **unattend.txt** enthalten, die gegebenenfalls modifiziert werden kann. Weitere Einzelheiten finden Sie unter http://www.microsoft.com/germany/ms/technetdatenbank/default.asp bzw. http://www.microsoft.com/TechNet/prodtechnol/windows2000serv/reskit/deploy/part7/append-c.asp (englischsprachig). Neben der Vorlagenantwortdatei haben Sie auch die Möglichkeit, den Windows 2000-Installations-Manager zum Erstellen einer Antwortdatei zu verwenden. Der Installations-Manager befindet sich auf der Windows 2000-CD im Ordner **\Support\Tools** in der Datei **deploy.cab**.

Unbeaufsichtigte Installation mit einer Netzwerkfreigabe

Wie bei der unbeaufsichtigten Installation mithilfe einer CD ist auch bei der unbeaufsichtigten netzwerkbasierten Installation die Verwendung von Antwortdateien erforderlich, um die Installation zu automatisieren. Sie können (wie bereits oben erwähnt) entweder die auf der Windows 2000-CD verfügbare Vorlagendatei **unattend.txt** oder den Windows 2000-Installations-Manager verwenden.

Den Unterschied zwischen der unbeaufsichtigten netzwerkbasierten Installation und der CD-basierten Installation macht das anfangs zum Starten der Installation erforderliche Medium aus. Im Falle einer netzwerkbasierten Installation wird die Installation mithilfe einer vorkonfigurierten Startdiskette gestartet.

Empfohlene Vorgehensweise – Einer der Gründe, aus denen sich die Kunden möglicherweise für diese Installationsmethode entscheiden, ist der Wunsch nach einer strengeren Kontrolle der Windows 2000-Installationsdateien. Wie bei allen Installationsdateien, die sich in einem einzigen Netzwerkverzeichnis befinden, müssen sich Administratoren keine Gedanken über möglicherweise falsche Dateiversionen machen.

Installation eines Abbildes mit Sysprep

Die vermutlich schnellste Methode, einen Computer unter Windows 2000 Server zu installieren und konfigurieren, besteht darin, eine Abbilderstellungssoftware zum Erstellen eines Serverabbildes sowie Sysprep zum Anpassen der Installation zu verwenden und dem Server eine eindeutige Sicherheits-ID (Security ID oder SID) zuzuweisen. Symantec Ghost™ und Powerquest DriveImagePro™ stellen zwei beliebte Möglichkeiten für Abbilderstellungssoftware dar.

Mithilfe dieser Methode wird ein Server durch Installieren von Windows 2000 und aller sonstigen erforderlichen Anwendungen erstellt. Anschließend wird *Sysprep* mit den entsprechenden Befehlszeilenoptionen ausgeführt, um den Server für die Duplizierung zu konfigurieren. Bedenken Sie, dass *Sysprep* das Originalsystem mithilfe der Medienduplizierungssoftware eines Drittanbieters, wie z. B. Ghost oder DriveImage, dupliziert. Schließlich wird nach dem Starten des neuen Systems eine neue Sicherheits-ID (SID) für diesen Server erstellt, und ein weiterer Installations-Assistent wird angezeigt. Dieser abschließende Installations-Assistent kann mithilfe der Datei **sysprep.inf** automatisiert werden.

Anmerkung Wenn Sie diese Installationsmethode bei Servern anwenden, die unterschiedliche Speichergeräte verwenden, sollten Sie Sysprep 1.1 verwenden, da Sie nur ein einziges Abbild verwalten müssen. Sysprep 1.1 unterstützt die Fähigkeit, mehrere Treiber für Massenspeichergeräte in einem einzigen Serverabbild einzuschließen. Dadurch wird die zum Erstellen der einzelnen Serverabbilder erforderliche Zeit sowie die Komplexität beim Verwalten mehrerer Serverabbilder reduziert.

Weitere Informationen zu Sysprep 1.1 finden Sie unter http://www.microsoft.com/windows2000/techinfo/planning/incremental/sysprep11.asp (englischsprachig)

Ermitteln der Einschränkungen jeder einzelnen Installationsmethode

Wie bereits oben erwähnt, hat jede Installationsmethode ihre Vor- und Nachteile. Es ist wichtig, dass der Systemadministrator sämtliche Einschränkungen in der spezifischen Umgebung ermittelt, bevor er sich für eine bestimmte Installationsmethode entscheidet. Möglicherweise ist es in Ihrer Umgebung sinnvoll, mehrere Installationsmethoden zu verwenden, und zwar netzwerkbasierte Installationen im Hauptbüro und CD-basierte Installationen in den Remoteniederlassungen ohne Netzwerkzugriff.

Im Folgenden sind einige Fragen aufgeführt, die Sie vor dem Auswählen einer Installationsmethode beantworten sollten:

- Liegen Einschränkungen aufgrund der Netzwerkbandbreite vor?
- Sind Installationen in Bereichen ohne Netzwerkzugriff erforderlich?
- Wird der Betriebssystem-Versionskontrolle eine hohe Priorität in Ihrer Umgebung beigemessen?
- Sind direkt Serveroperatoren zur Unterstützung zur Stelle, wenn eine Installation fehlerhaft ist?
- Sollen nur ganz bestimmte Personen in der Lage sein, das Betriebssystem zu installieren?
- Über welche Kenntnisse muss das Installationsteam verfügen? (Skripterstellung usw.)

Schritt 4: Ermitteln einer gegebenenfalls erforderlichen Konfiguration nach der Installation

Im Anschluss an eine Betriebssysteminstallation, die nicht mithilfe von Sysprep automatisiert werden kann, sind häufig aufgrund der jeweiligen Umgebung oder aufgrund der anzuwendenden Serverrolle spezifische Konfigurationsänderungen erforderlich.

Empfohlene Vorgehensweise – In der Regel erfolgt die folgende Konfiguration vor dem Installieren eines Servers in der Produktionsumgebung:

1. Installieren Sie das aktuelle Windows 2000 Service Pack
2. Installieren Sie entsprechende Hotfixes und Sicherheitsaktualisierungen
3. Deaktivieren Sie die Dienste, die für eine ordnungsgemäße Funktionsweise der Webanwendung nicht von Bedeutung sind
4. Stellen Sie sicher, dass die Ereignisprotokolleinstellungen korrekt sind
5. Fügen Sie den Server zu einer Domäne hinzu
6. Ändern Sie das Administratorkennwort
7. Netzwerkkonfiguration (Anpassen der NIC-Bindungen usw.)
8. Sicherheitskonfiguration (Erstellen von Benutzerkonten und NTFS-Partitionen)
9. Legen Sie IIS-Konfigurationsparameter fest

Schritt 5: Dokumentieren einer konsistenten, wiederholbaren Installationsmethode

Nach dem Auswählen einer Installationsmethode besteht der nächste Schritt darin, die für eine erfolgreiche Installation zu erfüllenden Anforderungen zu dokumentieren.

Empfohlene Vorgehensweise – Diese Anforderungen sollten zumindest ein Flussdiagramm enthalten, das auf jede im Laufe des Installationsprozesses durchgeführte Aktion sowie auf Maßnahmen hinweist, die im Falle eines Misserfolgs ergriffen werden sollten. Dies ist vor allem hilfreich, wenn Sie zukünftig die Probleme einer fehlerhaften Installation behandeln oder Ihre Installationsrichtlinien überarbeiten müssen. Mithilfe dieser Vorgehensweise war ein Dotcom-Unternehmen, mit dem Microsoft kooperiert, in der Lage, einen Prozess zu definieren, mit dem ein neuer Windows 2000-Webserver anhand eines Abbildes in weniger als 6 Minuten installiert werden kann. Diese Effizienz ist nicht nur für die Geschwindigkeit der Erstinstallation, sondern auch für Wiederherstellungsmaßnahmen wichtig, bei denen Zeit ein entscheidender Faktor ist.

Testen von Webservern

Es ist von großer Bedeutung, dass Sie die Webanwendung und die Webserver in einer kontrollierten Umgebung testen, bevor Sie die Anwendung der tatsächlichen Auslastung durch das Internet aussetzen. Beim Testen stehen zwei Ziele im Vordergrund. Zunächst können Sie Probleme mit der Hardware und der Webanwendung einfacher ausfindig machen. Zweitens wird dadurch Ihre Produktionsumgebung vor zufälligen Problemen geschützt und isoliert, die möglicherweise bei der neuen Anwendung auftreten könnten, während Sie die Tests durchführen. Vor dem Erstellen einer Testumgebung für Webanwendungen haben mehrere Dotcom-Kunden von Microsoft herausgefunden, dass bereits Anwendungskompilierungsfehler und Arbeitsspeicherverluste aufgetreten sind, nachdem sie ihre Webanwendung im Internet freigegeben hatten. Diese Probleme stellten erhebliche Ursachen der Ausfallzeiten und Kosten für diese Kunden dar. Sie haben herausgefunden, dass sie nach der Einführung eines formalen Anwendungstestprozesses in der Lage waren, diese Art von Problemen in ihrer jeweiligen Testumgebung nachzuvollziehen und zu lösen, was schließlich zu einer erhöhten Kundenzufriedenheit und geringeren Kosten führte.

Empfohlene Vorgehensweise – Richten Sie ein überwachtes Testlabor ein, um den Anwendungsentwurf und die Serverhardware zu testen, während Sie gleichzeitig Ihre Produktionsumgebung schützen. Der unten angegebene Link enthält detaillierte Informationen zum Einrichten eines Windows 2000-Testlabors, wie z. B. Informationen zum Beurteilen der Testmethodologie, zum Ermitteln der Voraussetzungen für das Entwerfen eines Labors sowie zur Clientcomputerkonfiguration.

- Erstellen eines Testlabors für Windows 2000 bzw. Building a Windows 2000 Test Lab (englischsprachig)

Das Durchführen von Belastungstests spielt auch eine wichtige Rolle, um sicherzustellen, dass die Anwendung bereitgestellt werden kann. Tools, wie z. B. Microsoft Web Application Stress Tool (WAS), ermöglichen ein problemloses Erstellen automatisierter Testskripts auf der Basis von Verwendungszielen, die Ihre Anwendung erfüllen soll. Beim Erstellen der Testskripts ist es wichtig, dass alle Komponenten der Webanwendung unter Belastung getestet werden. Sie sollten Skripts erstellen, um sowohl statische Inhalte als auch ASP, SSL usw. unter Belastung zu testen. Nachdem Sie mit den Belastungstests begonnen haben, ist es wichtig, den Webserver mithilfe des Systemmonitors zu überwachen (klicken Sie dazu auf **Start**, zeigen Sie auf **Programme**, klicken Sie dann auf **Verwaltung** und anschließend auf **Systemmonitor**), um zu ermitteln, ob die von Ihnen festgelegten Verwendungsziele in Anbetracht der Anwendungs- und Serverkonfiguration realisierbar sind. Detaillierte Empfehlungen zu den Leistungsindikatoren, die während des Testens überwacht werden sollten, finden Sie weiter unten im Abschnitt "Wichtige Leistungsindikatoren von Windows 2000".

Empfohlene Vorgehensweise – Verwenden Sie Web Application Stress Tool (WAS), um Belastungstests Ihrer Anwendung durchzuführen und künftige Entscheidungen hinsichtlich der Kapazitätsplanung leichter treffen zu können. Sie sollten sich vor dem Testen ein Ziel hinsichtlich der Belastung setzen, auf das Sie dann hinarbeiten können. Sie könnten beispielsweise Ihre Anwendungsentwickler vor die Herausforderung stellen, die Webanwendung ein bis zwei Wochen lang unter Belastung auszuführen, ohne dass Arbeitsspeicherverluste oder Anwendungsabstürze auftreten. Es ist zwar durchaus sinnvoll, die Verwendbarkeit der Site zu testen, darüber hinaus hat Microsoft jedoch festgestellt, dass das Ausführen von Belastungs- und Auslastungstests für sämtliche Komponenten der Anwendung letztendlich zu einer besseren Verfügbarkeit führt. Belastungstests unterstützen Sie nicht nur beim Identifizieren und Korrigieren unzureichender oder ineffizienter Codierungen in der Anwendung, sondern sie helfen Ihnen auch dabei, klügere Entscheidungen im Hinblick auf zukünftige Skalierbarkeitsanforderungen zu treffen. Weitere Informationen zum Konfigurieren von WAS finden Sie unter:

- Microsoft Web Application Stress Tool (englischsprachig)

Leistungsoptimierung von Webservern

Neben dem Verwenden der Testumgebung zum Suchen nach Codierungsfehlern in der Anwendung sollten Sie außerdem Tests durchführen, um mögliche Leistungsengpässe auf den Servern einfacher identifizieren und schließlich optimieren zu können. Das Optimieren der Webserver hat viele Vorteile. So kann beispielsweise durch geringere Serverwartezeiten ein besseres Umfeld für die Benutzer geschaffen, die Nutzung der aktuell verfügbaren Hardware maximiert und die Finanzplanung der Kapazitätssteigerungen verbessert werden. An dieser Stelle sollte darauf hingewiesen werden, dass Leistungsfähigkeit und Verfügbarkeit ganz eindeutig in Beziehung zueinander stehen. Es kann sich z. B. herausstellen, dass Symptome, wie eine langsame Anwendung, in vielen Fällen auf einen Arbeitsspeicherverlust zurückzuführen sind, der wiederum bei Nichtbeachtung sämtliche Systemressourcen verbrauchen und einen Serverausfall verursachen kann.

Microsoft hat ermittelt, dass die folgenden vier wichtigen Ressourcen streng überwacht werden sollten, um mögliche Engpässe auf den Webservern bestimmen zu können: Prozessorauslastung, Datenträger-E/A, Netzwerk-E/A und Speicherauslastung. Aus Messungen geht hervor, dass bei sehr hoher Auslastung dieser Ressourcen über einen längeren Zeitraum hinweg entweder weitere Ressourcen hinzugefügt oder die vorhandenen Ressourcen optimiert werden müssen.

Es muss unbedingt betont werden, dass die Leistungsüberwachung der Anwendung und Server in sämtlichen Phasen des Bereitstellungsprozesses erfolgen sollte. Sie sollten in der Testumgebung mit der Überwachung der Leistung beginnen und die Überwachung während der Bereitstellung bis hin zur Produktionsumgebung fortsetzen, um die möglicherweise auf den Systemen vorhandenen Engpässe erkennen zu können. Engpässe können aufgrund nicht ordnungsgemäß konfigurierter Hardware oder Software auftreten oder schlimmstenfalls aufgrund unzulänglicher Hardware, die der ausgesetzten Belastung nicht standhalten kann. Glücklicherweise stellt Microsoft jedoch Tools bereit, die Sie beim Messen der Leistungsfähigkeit unterschiedlicher Aspekte des Servers unter Windows unterstützen, von der Hardwareüberwachung bis zur Webanwendungsüberwachung. Verwenden Sie den in Abbildung 1 dargestellten Systemmonitor, um Leistungsengpässe einfacher identifizieren und nachvollziehen zu können, die sich möglicherweise nachteilig auf den Server unter Windows 2000 auswirken. Auf den Systemmonitor können Sie zugreifen, indem Sie im Startmenü auf **Programme** zeigen, dann auf **Verwaltung** und anschließend auf **Systemmonitor (Performance)** klicken.

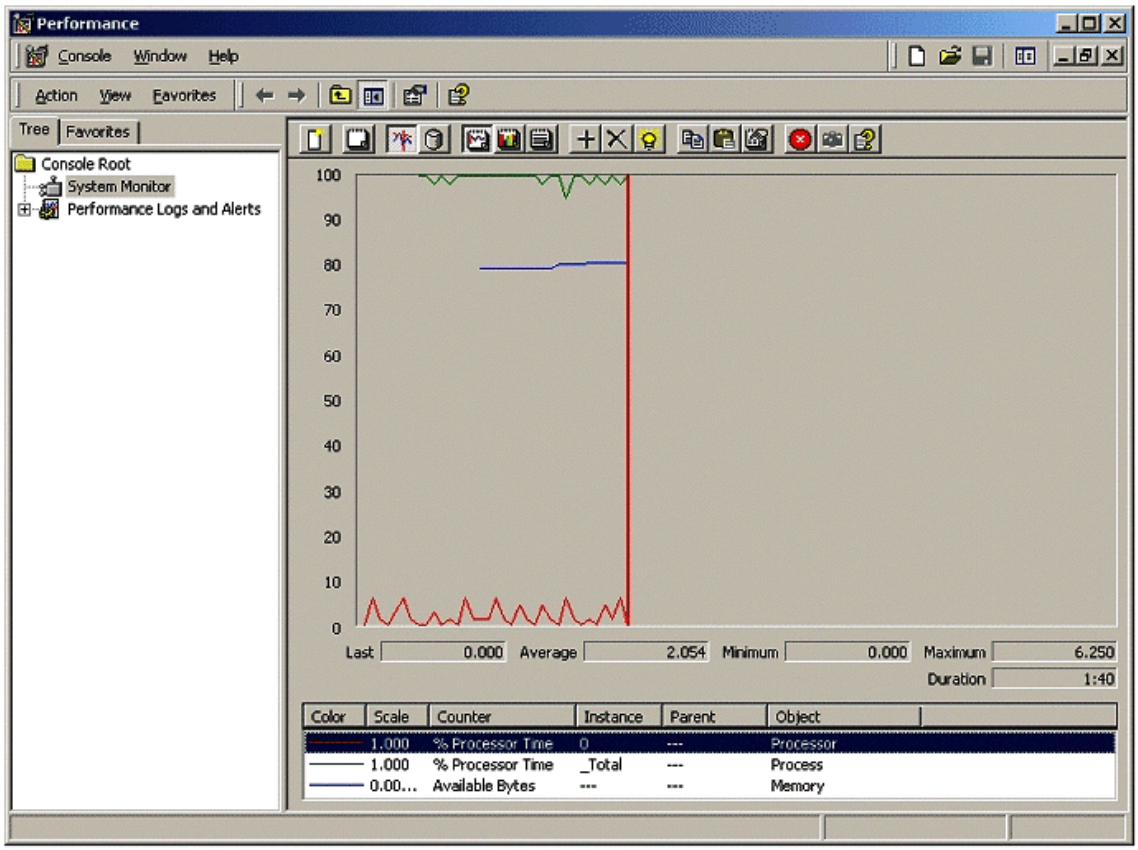


Abbildung 1: Systemmonitor

Der Systemmonitor gruppiert seine Inhalte nach Objekten, Leistungsindikatoren und Instanzen. Objekte bilden die höchste Abstraktionsebene und werden als logische Sammlung von Leistungsindikatoren definiert, die mit einer Ressource oder einem Dienst verknüpft ist, der überwacht werden kann. Bei Leistungsindikatoren handelt es sich um Datenelemente, die mit einem Leistungsobjekt verknüpft sind. Der Systemmonitor stellt für jeden ausgewählten Leistungsindikator einen Wert dar, der einem bestimmten, für dieses Objekt definierten Leistungsaspekt entspricht. Instanzen bilden die niedrigste Abstraktionsebene, und sie werden verwendet, um mehrere Leistungsobjekte desselben Typs auf einem Computer voneinander zu unterscheiden.

Wichtige Leistungsindikatoren von Windows 2000

Bevor Sie die Webanwendung im Hinblick auf Leistungsengpässe untersuchen, müssen Sie sich zuerst mit den Leistungsproblemen auf Betriebssystemebene befassen. Ein Ressourcenproblem, wie mangelnder Arbeitsspeicher, kann häufig noch weitere Leistungsprobleme auf dem System verursachen. Sie sollten daher jeden Bereich des Systems auf potenzielle Engpässe untersuchen, um die Systemleistung nachvollziehen und letztendlich verbessern zu können.

Überprüfen Sie die folgenden Leistungsindikatoren ganz genau, um ermitteln zu können, ob der Server Prozessor-, Arbeitsspeicher-, Datenträger- oder Netzwerkengpässe aufweist. Zusätzlich zur Beschreibung der einzelnen Leistungsindikatoren helfen Ihnen gemessene Grenzwerte und Empfehlungen dabei, die Leistungsengpässe auf den Webservern zu lösen. Sämtliche Messungen wurden auf Webservern mit 2 Prozessoren und 1 GB physischen RAM durchgeführt. Die Leistungsindikatoren werden gegebenenfalls im Format **Objekt:Indikator:Instanz** dargestellt.

Speicher: Verfügbare Bytes – Dieser Leistungsindikator zeigt den für die auf dem Computer ausgeführten Prozesse verfügbaren physischen Arbeitsspeicher in Bytes an. Dieser Wert wird durch Addieren des in den genullten, freien und Standby-Speicherlisten aufgeführten Speicherplatzes berechnet. Der freie Speicher kann verwendet werden. Genullter Speicher besteht aus mit Nullen aufgefüllten Speicherseiten, die verhindern, dass spätere Prozesse auf Daten von vorherigen Prozessen zugreifen können. Standbyspeicher ist Speicher, der noch verfügbar ist, aber aus dem Arbeitssatz eines Prozesses auf dem Weg zur Festplatte entfernt wurde. Dieser Indikator zeigt nur den letzten Wert an.

Niedrige Werte unter **Verfügbare Bytes** (4 MB oder weniger) können auf einen Speichermangel im gesamten Computersystem oder darauf hinweisen, dass ein Programm keinen Arbeitsspeicher freigibt. Bei Erreichen eines Grenzwertes von 4 MB wird das Ereignis 26 im Systemereignisprotokoll aufgezeichnet, um darauf hinzuweisen, dass das System nur über wenig virtuellen Arbeitsspeicher verfügt. Neben dem protokollierten Ereignis zeigt sich Speichermangel auch in einer allgemein langsamen Leistung eines betroffenen Systems.

Zum Vermeiden speicherspezifischer Leistungsprobleme sollten Sie mindestens 10 % des Arbeitsspeichers für die Spitzenlast der Website reservieren.

Speicher: Cachebytes – Dieser Indikator gibt die Größe des Dateisystemcaches an. Dieser Cache verwendet standardmäßig bis zu 50 % des verfügbaren physischen Arbeitsspeichers, und zwar bis maximal 900 MB, je nach Volumen des installierten physischen Arbeitsspeichers. Es handelt sich hierbei um eine wichtige Messgröße, da IIS im Falle eines Arbeitsspeichermangels den Cache automatisch verkleinert. Dieser Indikator kann zusammen mit dem unten aufgeführten Indikator **Private Bytes** verwendet werden, um eine Anwendung mit einem Arbeitsspeicherverlust zu isolieren.

Speicher: Seitenfehler/s– Mit diesem Indikator wird die durchschnittliche Anzahl von Seitenfehlern pro Sekunde gemessen. Dieser Indikator umfasst sowohl Hardwarefehler (die einen Datenträgerzugriff erfordern) als auch Softwarefehler (bei denen der Seitenfehler an einer anderen Stelle im physischen Arbeitsspeicher gefunden wird). Die meisten Prozessoren können eine große Anzahl von Softwarefehlern ohne wesentliche Folgen bewältigen. Hardwarefehler, die einen Datenträgerzugriff erfordern, können jedoch erhebliche Leistungsprobleme verursachen. Wenn die Werte für diesen Indikator niedrig sind, kann der Server schnell auf Anforderungen reagieren. Bei hohen Werten haben Sie möglicherweise zu viel Arbeitsspeicher für den Dateisystemcache reserviert, so dass nicht genügend Arbeitsspeicher für die anderen Systembereiche vorhanden ist. Ein dauerhaft hoher Wert der Seitenfehler pro Sekunde kann auf die Notwendigkeit hinweisen, den physischen Arbeitsspeicher auf dem Server zu vergrößern.

Prozess: Private Bytes: (Inetinfo, Dllhost) – Dieser Indikator misst den aktuellen von Inetinfo reservierten Arbeitsspeicher, der nicht gemeinsam mit anderen Prozessen verwendet werden kann. Das Überwachen dieses Indikators ist beim Isolieren von Arbeitsspeicherverlusten in der Webanwendung hilfreich. Sie werden feststellen, dass diese Prozesse mehr Arbeitsspeicher reservieren, ohne den Arbeitsspeicher über einen längeren Zeitraum hinweg wieder für das System freizugeben. Wenn für die Anwendung eine niedrige Isolationsstufe festgelegt ist, sollten Sie den Indikator **Private Bytes** für den Inetinfo-Prozess überwachen. Bei einer mittleren oder hohen Isolationsstufe sollten Sie die entsprechenden DLLHost-Prozesse überwachen.

System: Prozessor-Warteschlangenlänge – Dieser Indikator zeigt die Anzahl von Threads an, die auf die Ausführung in der Prozessorwarteschlange warten. Auch auf Computern mit mehreren Prozessoren ist nur eine Warteschlange für die Prozessorzeit vorhanden. Dieser Leistungsindikator erfasst nur abgeschlossene Threads, jedoch keine Threads, die noch ausgeführt werden. Eine andauernde Warteschlangenlänge von über zwei Threads pro Prozessor kann auf eine Prozessorüberlastung hinweisen, die sich dem Benutzer in Form eines langsamen oder nicht reagierenden Servers zeigt. Wenn mehrere Programmabläufe um einen Großteil der Prozessorzeit konkurrieren, kann der Durchsatz durch die Installation eines schnelleren Prozessors erhöht werden. Ein zusätzlicher Prozessor kann beim Ausführen von Multithreadprozessen hilfreich sein. Sie sollten dabei jedoch berücksichtigen, dass das Skalieren auf zusätzliche Prozessoren nur von begrenztem Vorteil sein kann.

Anmerkung Bei langen Threadwartezeiten kann die Prozessorwarteschlange bei einem Systemleerlauf größer als **2** sein.

Prozessor: Prozessorzeit (%) – Dieser Leistungsindikator zeigt eine prozentuale Angabe der Prozessorzeit an, die zum Ausführen eines Threads benötigt wird, der sich nicht im Leerlauf befindet. Dieser Leistungsindikator ist die primäre Anzeige der Prozessoraktivität und kann beispielsweise für die Problembehandlung bei einem langsamen oder nicht reagierenden Server verwendet werden. Es ist von großer Bedeutung, die Rolle des Computers beim Analysieren dieses Leistungsindikators zu verstehen. Wenn Sie z. B. den Computer eines Benutzers überwachen, der in erster Linie für eine CAD-Anwendung verwendet wird, kann die CAD-Anwendung während der Ausführung problemlos 100 % der Prozessorzeit in Anspruch nehmen. Auf einem Server, der viele Clientanforderungen verarbeitet, sind Werte um 100 % ein Hinweis darauf, dass die Prozesse in einer Warteschlange auf verfügbare Prozessorzeit warten und somit einen Engpass verursachen. Eine anhaltende Prozessorauslastung um 100 % ist daher auf Servern nicht akzeptabel und legt die Notwendigkeit zusätzlicher Prozessoren oder das Ändern (Aufteilen) der Arbeitsauslastung nahe. Ein akzeptabler Grenzwert für die Prozessorauslastung auf Webservern ist 70 %.

Netzwerkschnittstelle: Gesamtanzahl Bytes/s – Dieser Leistungsindikator misst die Rate, mit der Bytes über eine Netzwerkschnittstelle gesendet und empfangen werden. Wenn Sie ermitteln möchten, ob die Netzwerkverbindung einen Engpass verursacht, dann vergleichen Sie den Leistungsindikator *Netzwerkschnittstelle: Gesamtanzahl Bytes/s* mit der Gesamtbandbreite der Netzwerkkarte. Sie sollten im Allgemeinen nicht mehr als 50 % der Kapazität verwenden, um ausreichend Spielraum für Spitzen im Datenverkehr zu ermöglichen. Wenn dieser Wert nahezu der Kapazität der Verbindung entspricht und die Verwendung von Prozessor und Arbeitsspeicher normal ist, könnte die Verbindung das eigentliche Problem darstellen.

Physikalischer Datenträger: Zeit (%) – Dieser Leistungsindikator gibt den Prozentanteil der verstrichenen Zeit an, die das ausgewählte Laufwerk mit dem Bearbeiten von Lese- und Schreibanforderungen beschäftigt ist. Wenn der Wert dieses Leistungsindikators hoch ist (über 90 %), sollten Sie den Leistungsindikator **Physikalischer Datenträger: Aktuelle Warteschlangenlänge** überprüfen, um festzustellen, wie viele Systemanforderungen auf den Datenträgerzugriff warten. Die Anzahl von wartenden E/A-Anforderungen sollte auf Dauer nicht das 1,5- bis 2-fache der Spindelanzahl übersteigen, aus der sich der physische Datenträger zusammensetzt. (Wenden Sie sich an den Serverhersteller, um Informationen zu Spindeln zu erhalten.) Im Allgemeinen verfügen die meisten Datenträger über eine Spindel. RAID-Geräte verfügen jedoch in der Regel über mehrere Spindeln. Ein RAID-Hardwaregerät wird im Systemmonitor als einzelner physischer Datenträger angezeigt, während mithilfe von Software erstellte RAID-Geräte als mehrere Laufwerke angezeigt werden. Sie können wahlweise die Leistungsindikatoren **Physikalischer Datenträger** für jedes einzelne physische Laufwerk überprüfen, bei dem es sich nicht um ein RAID-Laufwerk handelt, oder die Instanz **_Total** verwenden, um die Daten sämtlicher Laufwerke des Computers zu überprüfen.

Verwenden Sie die Werte der Leistungsindikatoren **Physikalischer Datenträger: Aktuelle Warteschlangenlänge** und **Physikalischer Datenträger: Zeit (%)**, um Engpässe des Datenträgersubsystems zu erkennen. Wenn konstant hohe Werte vorliegen, sollten Sie das Laufwerk aktualisieren oder einige häufig verwendete Dateien auf einen anderen Datenträger oder Server verschieben.

Anmerkung Wenn Sie ein RAID-Gerät verwenden, kann der Leistungsindikator **Zeit (%)** einen Wert von über 100 % angeben. Wenn dies zutrifft, sollten Sie die Verwendung des Leistungsindikators **Physikalischer Datenträger: Durchschnittl. Warteschlangenlänge des Datenträgers** in Erwägung ziehen, um zu ermitteln, wie viele Systemanforderungen auf den Datenträgerzugriff warten.

Leistungsindikatoren für Webanwendungen

Zur Unterstützung der Administratoren beim Identifizieren von Problemen mit der Webanwendung hat Microsoft mehrere Objekte im Systemmonitor bereitgestellt. Verwenden Sie die unten aufgeführte Auflistung von Leistungsindikatoren für Webanwendungen, wenn Sie ermitteln möchten, ob Ihre Anwendung einen Leistungsengpass aufweist. Neben einer Beschreibung der einzelnen Leistungsindikatoren helfen Ihnen außerdem Grenzwerte und Empfehlungen beim schnellen Identifizieren und Lösen der Leistungsprobleme in der Anwendung. Sämtliche Grenzwerte und Empfehlungen basieren auf Messungen von Webservern mit zwei Prozessoren und 1 GB RAM. Die Leistungsindikatoren werden gegebenenfalls im Format **Objekt:Indikator:Instanz** dargestellt.

Active Server Pages: Ausgeführte Anforderungen – Dieser Leistungsindikator misst die Anzahl von aktuell ausgeführten Anforderungen. Er gibt an, ob die Anwendung jeweils eine Anforderung effektiv ausführt. Wenn der Wert für **Ausgeführte Anforderungen** nur **1** beträgt, werden die Anforderungen aus unbekannten Gründen serialisiert. Eine häufige Ursache für Serialisierung liegt vor, wenn Sie das ASP-Debuggen über den Internetdienste-Manager aktiviert haben.

Weitere Informationen und Codeserialisierungsbeispiele finden Sie unter:

- Blocking/Serialization When Using InProc Component (DLL) (englischsprachig)

Active Server Pages: Anforderungen in Warteschlange – Dieser Leistungsindikator misst die Anzahl von Anforderungen, die in der Warteschlange auf Bearbeitung warten (ideal wäre ein Wert von Null). Wenn diese Anzahl fortlaufend zunimmt, blockieren die ASPs Threads, und es werden keine Threads für die Verarbeitung einer weiteren Anforderung aus der Warteschlange freigegeben. Wenn die Anzahl von Anforderungen in der Warteschlange während eines Belastungstests beachtlichen Schwankungen unterliegt und die Prozessorauslastung relativ niedrig bleibt, ist dies ein Hinweis darauf, dass das Skript ein COM-Objekt aufruft, das wiederum mehr Aufrufe erhält als es verarbeiten kann. In diesem Fall stellt das COM-Objekt den Engpass dar. Zu diesem Zeitpunkt muss wahrscheinlich der Metabasiseintrag **ASPProcessorThreadMax** erhöht werden, der die maximale Anzahl der von IIS erstellten Arbeitsthreads pro Prozessor angibt.

Weitere Informationen zum Metabasiseintrag **ASPProcessorThreadMax** finden Sie unter:

- How to Tune the ASPProcessorThreadMax (englischsprachig)

Active Server Pages: Sitzungen insgesamt – Dieser Leistungsindikator misst die Gesamtanzahl von Sitzungen seit Dienststart. Wenn Sie die gesamten Sitzungen, die für ein bestimmtes Testskript erstellt werden, überwachen möchten, sollten Sie den Webdienst vor dem Testlauf beenden und starten, um die Gesamtanzahl genauer überwachen zu können. Stellen Sie sicher, dass die Anzahl von Sitzungen während der Skriptausführung allmählich zunimmt, bis die gewünschte Summe erreicht ist. Wenn die Gesamtanzahl von Sitzungen nicht die gewünschte Summe erreicht, möchten Sie möglicherweise den Webdienst beenden und neu starten und dann mit einem weiteren Testlauf beginnen.

WWW-Dienst: CGI-Anforderungen/s und ISAPI-Erweiterungsanforderungen/s – Diese Leistungsindikatoren messen jeweils die Geschwindigkeit, mit der Ihr Server CGI- und ISAPI-Anwendungsanforderungen verarbeitet. Wenn diese Werte bei zunehmender Belastung sinken, sollten die Anwendungsentwickler möglicherweise den Code überprüfen.

WWW-Dienst: GET-Anforderungen/s und POST-Anforderungen/s – Diese Leistungsindikatoren spiegeln die Geschwindigkeit wider, mit der diese zwei häufig auftretenden Typen von HTTP-Anforderungen an den Server gerichtet werden. POST-Anforderungen werden im Allgemeinen für Formulare verwendet und an ISAPIs (einschließlich ASP) oder CGIs gesendet. GET-Anforderungen machen fast alle anderen Anforderungen von Browsern aus und umfassen statische Dateien, Anforderungen für ASPs und andere ISAPIs sowie CGI-Anforderungen. Diese Leistungsindikatoren sind für das Verständnis der allgemeinen Auslastungsmerkmale der Website von Bedeutung.

Ausführlichere Informationen zur Webserveroptimierung finden Sie in den folgenden Quellen:

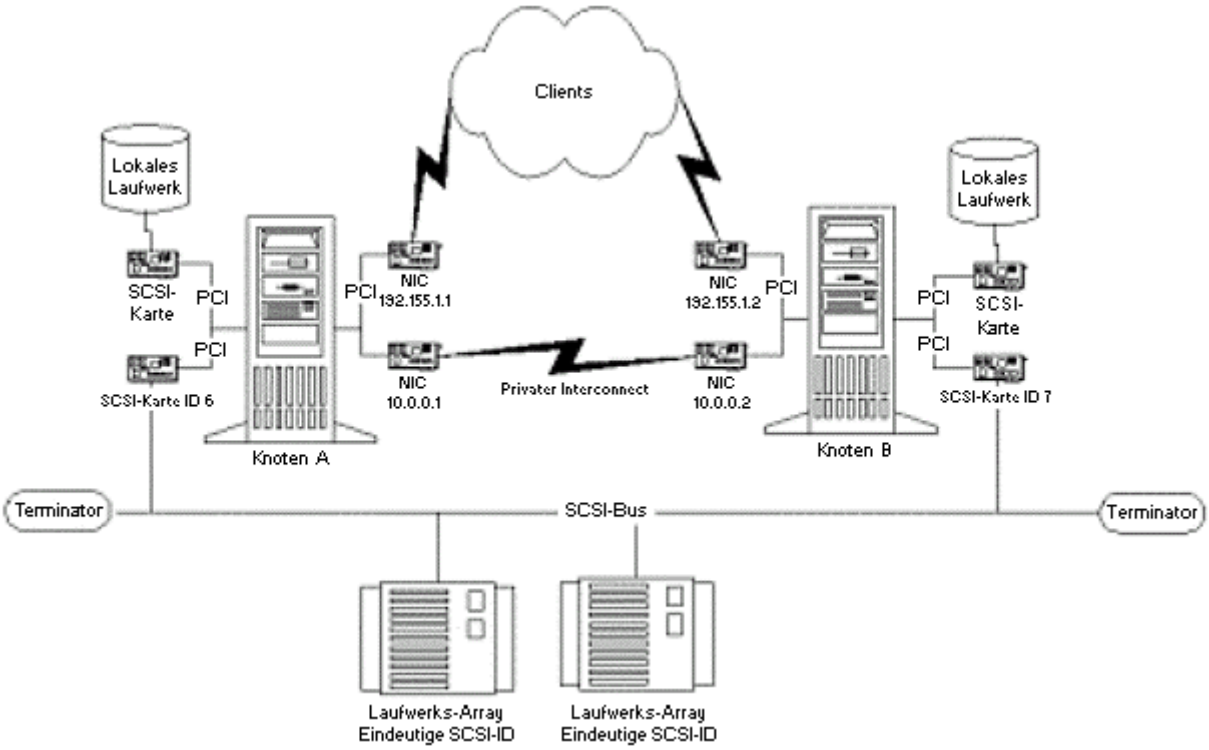
- Performance Management: Performance Monitor Counters to Monitor (englischsprachig)
- The Art and Science of Web Server Tuning with IIS5 (englischsprachig)
- Performance Monitor Counters for Web Server Testing (englischsprachig)
- Enable Logging for Process Accounting for IIS 5.0 (englischsprachig)

Erreichen einer Verfügbarkeit von 99,999 %

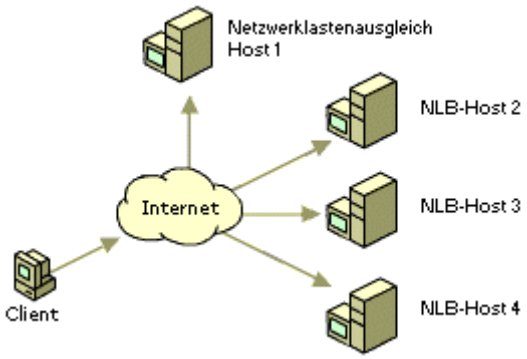
Wie bereits zuvor erwähnt, sollen die in diesem Dokument enthaltenen Informationen den Administrator darüber in Kenntnis setzen, wie ein Server mit einer Verfügbarkeit von 99,99 % entworfen, erstellt und verwaltet werden kann. Das Erreichen einer Verfügbarkeit von über 99,99 % stellt häufig ein sehr schwieriges Unterfangen dar. Glücklicherweise ist es jedoch möglich, die durch dieses Dokument erworbenen Kenntnisse einzusetzen, um eine Dienstverfügbarkeit von 99,999 % zu erreichen. Eine Dienstverfügbarkeit von 99,999 % bedeutet, dass die Zeit, in der der Webdienst nicht zur Verfügung steht, 5 Minuten pro Jahr nicht übersteigt. Dies ist zwar ein äußerst ehrgeiziges Ziel, es gibt jedoch praktikable Methoden, die Sie zum Erreichen dieses Zieles verwenden können.

Die effizienteste Strategie zum Erreichen einer Dienstverfügbarkeit von 99,999 % besteht darin, einen Netzwerklastenausgleich und eine Hardwarefehlertoleranz in den Webdienst zu integrieren. In der folgenden Liste werden unterschiedliche Komponenten detailliert aufgeführt, die Sie in Ihrer Umgebung zum Erreichen einer Dienstverfügbarkeit von 99,999 % einsetzen können.

- **Cluster.** Microsoft Cluster Server (MSCS) stellt eine Failoverunterstützung für wichtige Anwendungen und Dienste, wie z. B. SQL Server und Exchange, bereit. Microsoft Cluster Server eignet sich bestens für Serveranwendungen, die für die Unterstützung fehlertoleranter Transaktionen vorgesehen sind, die wiederum dauerhafte, serverbasierte Statusinformationen verwalten. Im Gegensatz zu NLB ordnet der Clusterdienst beim Auftreten eines Anwendungsfehlers bei MSCS die veröffentlichte IP-Adresse des virtuellen Servers einem verbleibenden Clusterknoten zu, so dass der Client die Verbindung wiederherstellen kann, ohne zu wissen, dass die Anwendung auf einem neuen physischen Knoten verwaltet wird. Windows 2000 Advanced Server unterstützt Cluster mit 2 Knoten, Windows 2000 Datacenter Server unterstützt Cluster mit 4 Knoten. In der folgenden Grafik wird die Architektur von MSCS zusammengefasst.



- **Netzwerklastenausgleich.** Der Microsoft-Netzwerklastenausgleich stellt eine Failoverunterstützung für TCP/IP-basierte Anwendungen und Dienste bereit, wie z. B. IIS und Windows Media-Dienste. Organisationen können den Netzwerklastenausgleich verwenden, um bis zu 32 Computer in einem Cluster zusammenzufassen, so dass sowohl der eingehende Datenverkehr gleichmäßig verteilt als auch die Fehlerfreiheit der Server und Netzwerkadapter überwacht werden können. NLB ist die ideale Lösung für unternehmenswichtige E-Commerce- und Hostanwendungen. Im Gegensatz zu MSCS sollte NLB nicht zum Skalieren von Anwendungen, wie z. B. Microsoft SQL Server, verwendet werden, die den clientübergreifenden Status unabhängig voneinander aktualisieren, weil die in einem Cluster vorgenommenen Aktualisierungen nicht für die anderen Clusterhosts sichtbar sind. NLB wird von Microsoft seit Windows NT 4 Enterprise Edition bereitgestellt, in der NLB als Windows-Lastenausgleichsdienst (Windows Load Balancing Service oder WLBS) bezeichnet wurde. In der folgenden Grafik wird die Architektur von NLB dargestellt.



Diese Funktion ist in Microsoft Application Center 2000 verfügbar und ermöglicht den COM+-Komponentenlastenausgleich. Beim Komponentenlastenausgleich (Component Load Balancing oder CLB) werden die COM+-Komponenten auf Servern in einem separaten COM+-Cluster verwaltet. Für die Aufrufe zum Aktivieren von COM+-Komponenten findet ein Lastenausgleich auf verschiedenen Servern innerhalb des COM+-Clusters statt. Weitere Einzelheiten zum COM+-Lastenausgleich und zu Application Center 2000 finden Sie im Artikel Technologieübersicht über den Komponentenlastenausgleich von Microsoft Application Center 2000 bzw. Application Center 2000 Component Load Balancing Technology Overview (englischsprachig).

- **Standbysysteme.** Standbysysteme stellen Sicherungssysteme bei einem Totalausfall des Hauptsystems bereit. MSCS kann diese Funktionalität bieten, wenn Sie den Cluster in einer Aktiv/Passiv-Konfiguration implementieren. Der passive Knoten wird nur verwendet, wenn eine Anwendung, wie SQL 2000 oder Exchange 2000, ein Failover erzwingt oder der Administrator ein Failover für den aktiven Knoten durchführt.
- **Ersatzteile.** Ersatzteile stellen sicher, dass im Falle eines Ausfalls immer entsprechender Ersatz verfügbar ist. Festplattenlaufwerke, Prozessoren und Arbeitsspeicher sind Beispiele für Ersatzteile, die Sie vor Ort bereitstellen können, um Hardwarefehler schnell zu beheben.
- **Fehlertolerante Komponenten.** Fehlertolerante Komponenten verbessern die interne Redundanz von Systemen. Speichergeräte, Netzwerkkomponenten, Lüfter und Stromversorgungen sind Beispiele für Komponenten, die für die Fehlertoleranz konfiguriert werden können. So können Speichergeräte z. B. konfiguriert werden, um eine Fehlertoleranz mithilfe von Redundant Array of Inexpensive Disks (RAID) bereitzustellen.

Empfohlene Vorgehensweise – Kooperieren Sie mit Ihrem Serveranbieter, um redundante Hardware für die Webserver einsetzen zu können, und verwenden Sie den Netzwerklastenausgleich-Dienst, um eine Dienstverfügbarkeit von 99,999 % zu erreichen. In den folgenden Artikeln werden die technischen Details von NLB, MSCS und CLB beschrieben, so dass Sie außerdem die mit jeder einzelnen Technologie verbundenen Vorteile nachvollziehen können.

- Technische Übersicht über den Netzwerklastenausgleich (Network Load Balancing) bzw. Network Load Balancing Technical Overview (englischsprachig)
- Windows-Clustertechnologien: Architektur des Clusterdienstes bzw. Windows Clustering Technologies: Cluster Service Architecture (englischsprachig)
- Technologieübersicht über den Komponentenlastenausgleich von Microsoft Application Center 2000 bzw. Application Center 2000 Component Load Balancing Technology Overview (englischsprachig)

Verwalten von Webservern

Nachdem Sie die Webanwendungen und Server sorgfältig im Hinblick auf Kompatibilität und Leistungsengpässe entworfen, erstellt und getestet haben, müssen Sie überlegen und planen, wie Sie diese Server überwachen und verwalten möchten. Bei jeder effizienten Strategie zum Verwalten und Überwachen von Webservern wird die zum Beurteilen und Nachvollziehen der Ausfallursachen erforderliche Zeit minimiert, und darüber hinaus wird sichergestellt, dass die Server die vorgesehene Leistung unter realen Auslastungsszenarien erbringen.

Verwaltungs- und Überwachungstools von Microsoft

Zu Beginn dieses Abschnitts werden die von Microsoft zur Verfügung gestellten und in Windows 2000 enthaltenen Verwaltungs- und Überwachungstools erläutert. Im nächsten Abschnitt werden dann die in Microsoft .Net Enterprise Server-Anwendungen bereitgestellten Tools beschrieben, die Sie zum Entwickeln einer Strategie für die Webserververwaltung verwenden können.

Verwalten und Überwachen mithilfe von im Lieferumfang von Windows 2000 enthaltenen Tools

Microsoft Windows 2000 Server-Produkte bieten Administratoren die Tools und Technologien, die sie für ein effizientes Verwalten von Servern und Netzwerkbenutzern von einem zentralen Standort aus benötigen. Neben dem bereits zuvor beschriebenen Systemmonitor sollten außerdem die folgenden Tools zur effizienten Verwaltung und Überwachung der Webserver verwendet werden.

- **Microsoft Management Console (MMC)** – MMC bietet eine konsistente Methode zum Durchführen von Verwaltungsaufgaben. Microsoft stellt eine Reihe von Snap-Ins bereit, die Administratoren zum Verwalten sämtlicher Bereiche der Server unter Windows 2000 verwenden können. So können Webserveradministratoren beispielsweise über das Snap-In **Internet-Informationsdienste** Verbindungen zu Remotewebservern herstellen und diese verwalten.

- **Windows Management Instrumentation (WMI)** – WMI ist ein Standard, mit dem Verwaltungsanwendungen unterschiedlicher Quellen die Geräte und Vorrichtungen einer Organisation, wie z. B. Dienste und Anwendungen, einheitlich verwalten können. Webserveradministratoren können mithilfe von WMI den Status von IIS-Diensten auf Remoteservern abfragen.
- **Terminaldienste** – Terminaldienste können im "Remoteverwaltungsmodus" installiert werden, der es Administratoren ermöglicht, zu Administrationszwecken (z. B. zum Starten und Beenden eines bestimmten Dienstes) von Remotecomputern eine Verbindung zur vertrauten Windows-Desktopumgebung herzustellen.
- **Taskplaner** – Der Taskplaner ermöglicht es, dass jedes Skript oder Programm in beliebigen Intervallen sowie für Ereignisse wie Systemstarts und Benutzeranmeldungen aufgerufen werden kann. Webserveradministratoren können den Taskplaner z. B. verwenden, um einen Task zum zeitlich geplanten Kopieren von IIS-Protokollen auf eine zentrale Verwaltungsstation zu erstellen.
- **Warndienst** – Der Warndienst erweitert die Funktionen des Systemmonitors dadurch, dass er es dem Administrator ermöglicht, Warnungen zu senden, wenn einer oder mehrere der Leistungsindikatoren einen vordefinierten Grenzwert übersteigen. Der Webserveradministrator kann z. B. konfigurieren, dass eine Warnung gesendet wird, wenn der Leistungsindikator **Prozessorzeit (%)** einen Wert von 70 % übersteigt.

Weitere Informationen zu diesen Tools und anderen Verwaltungsprogrammen, die im Lieferumfang von Windows 2000 enthalten sind, finden Sie in den folgenden Artikeln:

- Windows 2000 Management Features (englischsprachig)
- Windows 2000 Centralized Management (englischsprachig).

Verwalten und Überwachen mithilfe von .Net Enterprise Server-Anwendungen

Neben den in Windows 2000 enthaltenen Verwaltungs- und Überwachungstools hat Microsoft Verwaltungsanwendungen entwickelt, die Administratoren beim Überwachen und Verwalten der Ereignisse und Leistungsfähigkeit von Windows 2000-basierten Systemen unterstützen.

Microsoft Operations Manager 2000

Microsoft Operations Manager 2000 stellt ein Ereignis- und Leistungsmanagement für die Windows 2000 Server-Familie der Betriebssysteme und .Net Enterprise Server-Anwendungen bereit. Im Wesentlichen erfüllt diese Anwendung folgende Aufgaben:

- Bereitstellen eines umfassenden Ereignismanagements mithilfe eines Unternehmensereignisprotokolls, das Probleme und Informationen sammelt und meldet, die von den Systemen und Anwendungen des gesamten Netzwerks erzeugt werden.
- Bereitstellen von proaktiver Überwachung und Warnmeldungen für Pager per E-Mail oder über andere externe Hilfsmittel. Diese Warnungen können auch Aktionen hervorrufen, mit denen das ursprüngliche Problem behoben wird.
- Bereitstellen von Berichterstellung und Trendanalysen, die zum Nachverfolgen von Problemen über einen längeren Zeitraum hinweg sowie zum Erstellen detaillierter Berichte über den Allgemeinzustand der Umgebung verwendet werden können.
- Interoperabilität mit anderen Managementsystemen durch die Unterstützung für Managementtechnologiestandards, wie SNMP, WMI und CIM.

Microsoft Application Center 2000

Microsoft Application Center 2000 ist ein von Microsoft entwickeltes Bereitstellungs- und Verwaltungstool für Webanwendungen mit hoher Verfügbarkeit, die auf der Windows 2000-Plattform erstellt wurden. Application Center 2000 ermöglicht Webentwicklern und Administratoren das problemlose Bereitstellen von Anwendungen auf mehreren Webservern. Im Wesentlichen erfüllt diese Anwendung folgende Aufgaben:

- Sie ermöglicht Administratoren das Erstellen logischer Anwendungsgruppen, einschließlich der Komponenten und Konfigurationen, mit denen die Komplexität beim Installieren und Testen minimiert wird.
- Sie ermöglicht Administratoren das Verwalten einer Gruppe von Servern als Einheit.
- Optimieren der Anwendungsbereitstellung, mit der die Konsistenz von Anwendungen auf allen Webservern sichergestellt wird.
- Bereitstellen von Tools zum Überwachen von Webservern, die das Anzeigen von Leistungs- und Ereignisprotokolldaten eines Servers oder des gesamten Clusters über eine Remoteanzeige der Webbenutzeroberfläche ermöglichen.
- Unterstützung beim Beseitigen manueller Aufgaben mit der Fähigkeit, Reaktionen auf Ereignisse auf der Grundlage bestimmter Ereignisse und Bedingungen zu automatisieren

Weitere Einzelheiten zu diesen beiden Anwendungssuites finden Sie auf den folgenden Websites:

- Homepage zu Microsoft Operations Manager bzw. Microsoft Operations Manager Home Page (englischsprachig)
- Microsoft Application Center bzw. Microsoft Application Center (englischsprachig).

Nachdem Sie nun Kenntnisse in Bezug auf die von Microsoft angebotenen Verwaltungs- und Überwachungsprogramme haben, werden in den nächsten Abschnitten empfohlene Vorgehensweisen bei Serverausfällen und Wiederherstellungsmaßnahmen, wie beispielsweise das Debuggen von Windows 2000 und Ihrer Webanwendung mithilfe des IIS Recycle-Tools, sowie Empfehlungen für das Auswerten und Installieren von Sicherheitsaktualisierungen und Hotfixes erläutert.

Empfohlene Vorgehensweisen für Serverausfälle und Wiederherstellungsmaßnahmen

Neben der Leistungsüberwachung und dem Planen der Webserverbereitstellung müssen Sie außerdem über einen Plan im Falle eines Webserverausfalles verfügen. Es sind viele Tools von Microsoft verfügbar, mit denen Sie die Ursache für einen Ausfall ermitteln und den Webserver wieder in den normalen Betriebszustand zurückversetzen können. Wie bereits zuvor im Abschnitt über die Verfügbarkeit erwähnt, darf die Zeit der Ausfälle nicht mehr als 52 Minuten pro Jahr betragen, um eine Verfügbarkeit eines Einzelservers von 99,99 % zu erreichen. Zu diesem Zweck müssen Sie die mit Fehlerereignissen verbundene Ausfallzeit minimieren, indem Sie die Ursache schnell erkennen und sich bemühen, den Webserver rechtzeitig für die Aufnahme des Produktionsdienstes wiederherzustellen.

In den folgenden Abschnitten werden Tools und Prozesse umrissen, die Sie verwenden können, um die Ursachen für Fehler in Ihrer Umgebung einfacher identifizieren sowie das Auftreten von Ausfällen ganz vermeiden zu können.

Debuggen von Windows 2000

Windows 2000 umfasst viele Debugtools, die Sie zum Isolieren von Fehlern in einer Webanwendung verwenden können. Die Debugtools sind auf der im Lieferumfang von Windows 2000 Server und Windows 2000 Advanced Server enthaltenen Customer Support Diagnostics-CD zu finden. Diese CD umfasst herkömmliche Windows NT-Debugtools, wie i386kd, KD, Windbg und CDB, sowie die zum Verwenden der Debugtools erforderlichen Symbole. Diese Tools sind beim Lösen von Problemen, wie Zugriffsverletzungen und nicht reagierende Anwendungen, äußerst hilfreich.

Empfohlene Vorgehensweise – Downloaden und installieren Sie die aktuellen Debugtools, damit diese Sie beim Diagnostizieren und Lösen von Problemen mit der Webanwendung unterstützen:

- Windows 2000 Customer Support Diagnostics
- Microsoft Debugtools bzw. Microsoft Debugging Tools for Windows (englischsprachig).

Debuggen von Webanwendungen

Neben den oben erwähnten Standardtools zum Debuggen sind weitere Ressourcen zum Debuggen von Webanwendungen verfügbar. Diese Tools und Prozesse wurden in Folge eines direkten Kundenfeedbacks seitens vieler Dotcom-Unternehmen entwickelt.

VBCHKW2K

Nach dem Analysieren von Kundensicherungsdateien aus abgestürzten oder nicht reagierenden COM+-Anwendungen und Active Server Pages wurde häufig festgestellt, dass die erforderlichen Einstellungen für serverseitige Komponenten nicht vorhanden waren. Zwei der erforderlichen Einstellungen für serverseitige Komponenten umfassen das Festlegen der Optionen **In Speicher erhalten** und **Unbeaufsichtigte Ausführung**. Visual Basic 6.0-ActiveX-DLLs (Dynamic Link Libraries), die ohne diese Optionen kompiliert wurden, können den Absturz von COM+-Anwendungen und Websites verursachen. Standardmäßig sind die Optionen **In Speicher erhalten** und **Unbeaufsichtigte Ausführung** bei ActiveX-DLLs, die mithilfe von Microsoft Visual Basic 6.0 erstellt wurden, nicht aktiviert.

Bei VBCHKW2K handelt es sich um ein Tool, das nach VB6-DLLs sucht, die kompiliert wurden, *ohne* die Eigenschaften **In Speicher erhalten** und **Unbeaufsichtigte Ausführung** aktiviert zu haben.

- Q264957 BUG: VB DLL Instability in COM+ if 'Retained In Memory' Not Set (englischsprachig)
- Q241896: Threading Issues with Visual Basic 6.0 ActiveX Components (englischsprachig)

Die Dokumentation zum Tool VBCHKW2K finden Sie im Anhang. Zusätzlich zum Tool VBCHKW2K ist auch der folgende Link beim Debuggen von COM-Objekten in Windows 2000 hilfreich.

- How to Debug InProc COM Components Inside IIS Using Windbg (englischsprachig)

Verhindern von Webanwendungsfehlern

Selbst wenn Sie erheblich viel Zeit und Arbeit in das Entwerfen und Testen der Webanwendung investiert haben, werden Sie möglicherweise auf Probleme stoßen, nachdem Sie die Anwendung für die Produktionsumgebung freigegeben haben. Das Einbauen von Fehlern und die Belastungssimulation in einem Testlabor schaffen noch keine Umgebung, die tatsächlich mit dem Internet identisch ist.

IIS 5 Recycle-Tool

Microsoft hat kürzlich ein Dienstprogramm entwickelt, mit dem Sie IIS-Prozesse automatisch wieder verwenden können, wenn Sie Probleme mit Ihrer Anwendung feststellen, wie z. B. einen Arbeitsspeicherverlust nach dem Freigeben der Anwendung für die Produktionsumgebung. Standardmäßig kann dieses Tool IIS automatisch auf der Grundlage folgender Komponenten wieder verwenden: dem virtuellen Arbeitsspeicher, der insgesamt verarbeiteten HTTP-*GET-Anforderungen* (werden anhand des Leistungsindikators *GET-Anforderungen insgesamt* ermittelt), der geplanten, der IIS-Betriebszeit-basierten und der in Warteschlangen stehenden ASP-Anforderungen. IIS Recycle verwendet sowohl prozessinterne Webanwendungen (**Inetinfo.exe**) als auch prozessexterne Webanwendungen (**dllhost.exe**) mehrmals.

Das IIS Recycle-Tool verfügt über eine integrierte Unterstützung für NLB. Auf einem NLB-fähigen System entfernt das IIS Recycle-Tool den Webserver vor der Wiederverwendung des IIS-Prozesses aus dem NLB-Cluster. Im Anschluss an die Wiederverwendung des IIS-Prozesses fügt das IIS Recycle-Tool den Webserver wieder zum NLB-Cluster hinzu. Darüber hinaus hat Microsoft ein Wasserzeichen als Teil der NLB-Unterstützung in IIS Recycle integriert. Der Administrator kann ein Wasserzeichen oder die Mindestanzahl von aktiven Hosts in einem NLB-Cluster konfigurieren. Wenn die aktuelle Anzahl von aktiven Hosts unterhalb des Wasserzeichens liegt oder diesem entspricht, überspringt das IIS Recycle-Tool die Wiederverwendung, bis die Anzahl von aktiven Hosts den Grenzwert überschritten hat.

Das IIS Recycle-Tool umfasst auch die Fähigkeit, benutzerdefinierte Skripts vor und nach jeder IIS-Wiederverwendung auszuführen. Einige Kunden haben diese Skriptfunktion verwendet, um Leerungsanhaltungsvorgänge für Lastenausgleichlösungen von Drittanbietern zu aktivieren und Nachrichten an Administratoren zu senden, die darauf hinweisen, dass die IIS-Dienste auf einem bestimmten Webserver erfolgreich wieder verwendet wurden. Auf einem NLB-fähigen System werden die Skripts nach dem NLB-Ausgleich und vor dem NLB-Start ausgeführt.

In Abbildung 4 werden die Betriebszeitverbesserungen eines Produktionswebservers an einem Microsoft-Standort nach dem Installieren des IIS Recycle-Tools dargestellt. Vor der Installation des IIS Recycle-Tools wurden alle Webserver wöchentlich neu gestartet. Nach der Installation von IIS Recycle wurde kein Server neu gestartet, und wie in der folgenden Abbildung dargestellt wird, umfasst die Betriebszeit des Webservers nun mehr als 60 Tage.

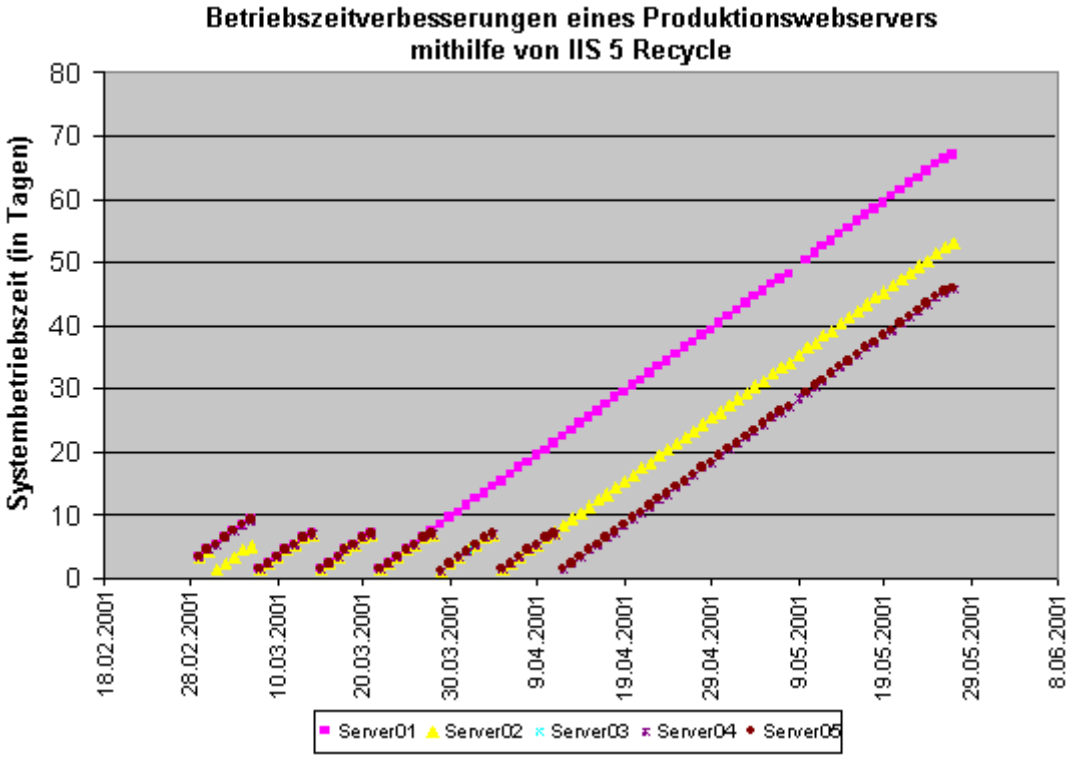


Abbildung 2: Betriebszeitverbesserungen eines Produktionswebservers mithilfe des IIS 5 Recycle-Tools

Empfohlene Vorgehensweise – Verwenden Sie das IIS 5 Recycle-Tool, um die Webanwendung wieder zu verwenden und künftige Anwendungsfehler über vom Administrator definierte Regeln zu vermeiden. Das IIS 5 Recycle-Tool kann mithilfe des folgenden Links vom Microsoft Download Center gedownloadet werden. In Zukunft wird das IIS 5 Recycle-Tool auch über MSDN, TechNet und das IIS Resource Kit verfügbar sein.

Downloaden Sie das IIS 5 Recycle-Tool (englischsprachig).

Die Dokumentation zum IIS 5 Recycle-Tool, einschließlich der benutzerdefinierten Konfigurationseinstellungen, steht im Anhang zur Verfügung.

Geplante Aktualisierungen

Neben der Leistungsüberwachung stellt das Verwalten von Softwarewartungen und Softwareaktualisierungen eine weitere große Herausforderung für jeden Systemadministrator dar. Die Softwarewartung kann mehrere Formen annehmen, beispielsweise das Anwenden von Hotfixes, Service Packs und Sicherheitsaktualisierungen. Jede Form kann eine andere Wartungsstrategie sowie gesonderte Installationsmechanismen umfassen. Für einen Systemadministrator ist es wichtig, dass er beurteilen kann, wie wichtig eine bestimmte Problemlösung für die jeweilige Umgebung ist. Darüber hinaus muss er die effizienteste Möglichkeit für das Installieren dieser bedeutenden Problemlösungen kennen.

Eine der Möglichkeiten, die Microsoft zum Warten der Server gewählt hat, besteht in der Einführung eines geplanten Plattformupdates, die auf allen von Microsoft verwalteten Servern installiert wird. Dieses Plattformupdate findet zweimal pro Jahr statt und enthält alle freigegebenen Service Packs, getestete Hotfixes und kann auch Softwareaktualisierungen zum Überwachen oder Verwaltungssoftware- und serverspezifische Hardwareupdates (Firmware, BIOS usw.) umfassen. In der Vergangenheit hat Microsoft Service Packs und Hotfixes getestet und bereitgestellt, sobald diese verfügbar waren. Dies hat sich zu einer komplexen Verwaltungsaufgabe entwickelt, da die Zeit zwischen den Service Pack-Versionen abnahm. Durch das Einhalten eines standardmäßig zweimal jährlich stattfindenden Plattformupdates kann Microsoft sicherstellen, dass seine Kunden (Mitarbeiter, Partner und Unternehmenseinheiten) eine supportfähige und zuverlässige Windows-Konfiguration ausführen können.

Sicherheitsaktualisierungen

Es ist zwar nicht immer erforderlich, jede verfügbare Sicherheitsaktualisierung zu installieren, dafür ist es aber auf jeden Fall wichtig, die Anwendbarkeit jeder einzelnen Sicherheitsaktualisierung in Ihrer Umgebung zu beurteilen. Wenn Sie z. B. Windows 2000 Active Directory nicht in Ihrer Organisation bereitstellen, werden Sie feststellen, dass das Microsoft Sicherheitsbulletin MS01-024 (englischsprachig) für Ihre Zwecke nicht zutrifft. Jede Sicherheitsaktualisierung enthält detaillierte Informationen, mit denen Sie ermitteln können, ob die entsprechende Aktualisierung für Ihre Umgebung geeignet ist. Wenn der Systemadministrator die Sicherheitsrisiken und die Plattformkonfiguration nachvollzogen hat, kann er einfacher entscheiden, welche bestimmte Sicherheitsaktualisierung installiert werden soll.

Empfohlene Vorgehensweise – Verwenden Sie das Tool HFNetCheck in regelmäßigen Abständen, um zu ermitteln, welche Microsoft-Sicherheitsaktualisierungen noch nicht für die Webserver übernommen wurden. HFNetChk ist ein Befehlszeilenprogramm, das es einem Administrator ermöglicht, den Patchstatus aller Computer in einem Netzwerk von einem zentralen Standort aus zu überprüfen. Das Tool bezieht sich dabei auf eine von Microsoft verwaltete XML-Datenbank. HFNetChk kann auf Windows NT 4.0- oder Windows 2000-Systemen ausgeführt werden. Das Tool durchsucht sowohl lokale als auch Remotesysteme auf Patches, die für Windows NT 4.0, Windows 2000, Internet Explorer 5.01 und höher, Microsoft SQL Server 7.0 und SQL Server 2000 sowie für alle Windows-Systemdienste einschließlich Internet Information Services 4.0 und Internet-Informationsdienste 5.0 verfügbar sind.

Im Folgenden erhalten Sie ein Beispiel für die Ausgabe, die nach dem Ausführen von HFNetCheck angezeigt wird:

```
C:\downloads\temp>hfnetchk -h server1
Microsoft Network Security Hotfix Checker, 3.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)
** Attempting to download the XML from
http://download.microsoft.com/download/xml/security/1.0/NT5/EN-
US/mssecure.cab. **
** File was successfully downloaded. **
** Attempting to load C:\downloads\temp\mssecure.xml. **
Using XML data version = 1.0.1.137 Last modified on 8/21/2001.
Scanning server1
.............
Done scanning server1
---------------------------
server1
---------------------------
WINDOWS 2000 ADVANCED SERVER SP2
Patch NOT Found MS00-077 Q299796
Patch NOT Found MS00-079 Q276471
Patch NOT Found MS01-007 Q285851
Patch NOT Found MS01-013 Q285156
Patch NOT Found MS01-025 Q296185
Patch NOT Found MS01-031 Q299553
Patch NOT Found MS01-036 Q299687
Patch NOT Found MS01-037 Q302755
Patch NOT Found MS01-040 Q292435
Patch NOT Found MS01-041 Q298012
Patch NOT Found MS01-046 Q252795
Internet Information Services 5.0
Patch NOT Found MS01-025 Q296185
Patch NOT Found MS01-044 Q301625
Internet Explorer 5.01 SP2
Patch NOT Found MS01-027 Q295106
```

Informationen zum Downloaden von HFNetCheck finden Sie unter http://support.microsoft.com/default.aspx?scid=kb;en-us;Q303215 (englischsprachig).

Hotfixverwaltung

Wie bei den Sicherheitsaktualisierungen muss auch nicht unbedingt jedes verfügbare Windows-Hotfix installiert werden. Hotfixes werden erstellt, um spezifische, reproduzierbare Fehler bei einer bestimmten Windows-Komponente zu beheben. Im Gegensatz zu Service Packs werden für Hotfixes keine umfassenden Regressionstests durchgeführt. Sie müssen demnach die Hotfixes vor dem Bereitstellen auf den Produktionsservern erst in einer Testumgebung testen.

Zusätzlich zu der zuvor beschriebenen halbjährlich geplanten Aktualisierung hat das Team des Microsoft-Rechenzentrums einen Standardprozess zum Auswerten der Anwendbarkeit und Priorität von freigegebenen Hotfixes und Sicherheitsaktualisierungen entwickelt und übernommen. Das folgende System wurde für die Festlegung einer Prioritätsreihenfolge entwickelt, um internen Kunden von Microsoft dabei behilflich zu sein, intelligente Entscheidungen im Hinblick auf das Installieren von Problemlösungen außerhalb der halbjährlich geplanten Aktualisierungen zu treffen:

1. **Wichtig** – Diese Problemlösung sollte so bald wie möglich auf allen Systemen installiert werden. Ein Beispiel hierfür ist eine Sicherheitsaktualisierung, die überprüft und für die Microsoft-Umgebung als geeignet befunden wurde.
2. **Dringend empfohlen** – Wird als erforderlicher Bestandteil der Standardplattform angesehen. Wenn das Debuggen eines Servers erfolgen soll, muss diese Problemlösung vor dem Debugvorgang installiert werden.
3. **Empfohlen** – Wird als erforderlicher Bestandteil der Standardplattform angesehen. Diese Problemlösung ist zwar nicht unbedingt für jeden Server erforderlich, sie hilft jedoch dabei, eine stabile Plattform auf allen Servern aufrechtzuerhalten.
4. **Eventuell erforderlich** – Diese Problemlösung wird zum Beheben ganz spezieller Problemfälle benötigt und ist wahrscheinlich nicht im nächsten geplanten Plattformupdate enthalten.
5. **Vermutlich nicht erforderlich** – Wird nur in ganz besonderen Fällen benötigt. Diese Problemlösung wird auf gar keinen Fall im nächsten geplanten Plattformupdate enthalten sein.

Das Microsoft Windows Sustained Engineering Team hat erheblich viel Arbeit investiert, um die Anzahl von beim Installieren von Hotfixes erforderlichen Neustarts zu reduzieren. Bei Hotfixes sowie Sicherheits- und wichtigen Aktualisierungen handelt es sich häufig um Problemlösungen für Benutzermoduskomponenten. In den meisten Fällen ist es möglich, einen Hotfix auf eine Benutzermoduskomponente anzuwenden und den Hotfix zu aktivieren, ohne das System erneu starten zu müssen. So besteht beispielsweise in bestimmten Situationen die Möglichkeit, einen ausgeführten Dienst zu beenden, die Dateien des Hotfixes auf das System zu kopieren und den Dienst erneut zu starten. Da die Dateien beim Beenden des Dienstes nicht gesperrt waren, konnten die alten Dateien problemlos mit den neuen Dateien überschrieben werden. Nach dem Neustart des Dienstes wurden die neuen Dateien verwendet, und der Hotfix wurde somit installiert. Diese Funktionalität wird in einer Aktualisierung des Microsoft Hotfix-Installationsprogramm enthalten sein, der im Anschluss an Windows 2000 Service Pack 2 freigegeben wird. Neben der zusätzlich im Hotfix-Installationsprogramm integrierten Unterstützung zum Beenden und Starten von notwendigen Diensten, bietet Microsoft auch zusätzliche Unterstützung für die Hotfixverkettung. Bei dieser Funktion handelt es sich um ein Tool, das Kunden verwenden können, um mehrere Hotfixes mit nur einem Neustart sicher miteinander zu verketten.

Verwenden von QFECheck zum Ermitteln installierter Hotfixes

Eine Herausforderung, mit der viele Systemadministratoren konfrontiert werden, stellt das Ermitteln der Betriebssystem- und Hotfixkonfiguration jedes einzelnen Servers in der Umgebung dar. Bis vor kurzem war dies oft eine schwierige Aufgabe, die eine benutzerdefinierte Entwicklung von Tools und Skripts erforderte, um unzusammenhängende Hotfixinstallationsinformationen aus der Registrierung und dem Dateisystem zu ziehen. Ein Tool mit der Bezeichnung QFECheck wurde in ein aktuelles Microsoft Sicherheitsbulletin integriert. Sinn und Zweck dieses Tools ist die Problembehebung beim Ermitteln der Hotfixes und der Betriebssystemversion eines Computers unter Windows 2000.

Empfohlene Vorgehensweise – Downloaden Sie QFECheck, und führen Sie das Programm aus, um die auf den Computern unter Windows 2000 installierten Hotfixes und Service Packs zu ermitteln. QFECheck kann auch zum Überprüfen der erfolgreichen Installation eines Windows 2000-Hotfixes (z. B. einer Sicherheitsaktualisierung) verwendet werden und bietet Anweisungen, wenn ein Konfigurationsproblem mit einem Hotfix auf dem Computer ermittelt wird.

Ein Beispiel für die Ausgabe von QFECheck kann folgendermaßen aussehen:

```
Windows 2000 Hotfix Validation Report for \\Server1
Report Date: 3/1/2001 11:27am
Current Service Pack Level: Service Pack 1
Hotfixes Identified:
Q259524: Current on system.
Q280838: Current on system.
Q282784: Current on system.
```

Die Remoteausführung von QFECheck für eine Gruppe von Servern ist nicht möglich. Es besteht jedoch die Möglichkeit, QFECheck über **Geplante Tasks** auszuführen und das Protokoll an einem einzigen Standort auszugeben, um so die Hotfixverwaltung zu vereinfachen.

Downloaden Sie QFECheck.

Anhang

Registrierungsskript zum Deaktivieren nicht benötigter Dienste

Im folgenden Beispiel wird dargelegt, wie viele Kunden sich für das Deaktivieren von Windows 2000-Diensten entschieden haben, die für die jeweilige Webanwendung nicht erforderlich sind. Bei der folgenden Liste handelt es sich nicht um eine Empfehlung von Microsoft, welche Dienste deaktiviert werden sollten, sondern es ist vielmehr eine Auflistung der Dienste, die viele Kunden deaktiviert haben.

```
REGEDIT4
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \AppMgmt]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \ClipSrv]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Fax]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Messenger]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \NetDDE]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Spooler]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \UPS]
"Start"=dword:00000004
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \W32Time]
"Start"=dword:00000004
```

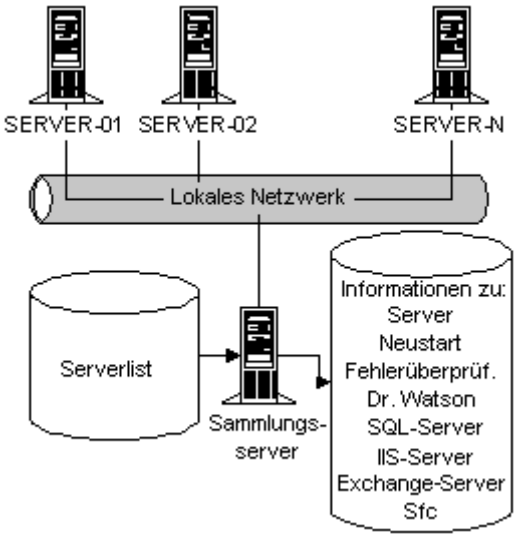
Registrierungsskript zum Vergrößern des Ereignisprotokolls

Neben der Verwendung von REG-Dateien zum Verwalten der Liste ausgeführter Dienste haben sich viele Kunden auch dafür entschieden, die REG-Dateien zu verwenden, um die ordnungsgemäße Konfiguration der Ereignisprotokolle sicherzustellen. Im Folgenden wird eine typische Methode aufgezeigt, die Kunden zum Konfigurieren von Ereignisprotokollen mithilfe der Registrierung auswählen:

```
Regedit4
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Eventlog
\Application]
"MaxSize"=dword:12320768
"Retention"=dword:00000000
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Eventlog
\Security]
"MaxSize"=dword:12320768
"Retention"=dword:00000000
[HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Eventlog \System]
"MaxSize"=dword:12320768
"Retention"=dword:00000000
```

Das Event Log Analyst-Tool

Bei Event Log Analyst (ELA) handelt es sich um ein Tool, das Informationen zur Zuverlässigkeit aus Windows NT-Ereignisprotokollen sammelt. ELA wird auf einem einzelnen Server (dem so genannten Sammlungsserver) ausgeführt und ruft fortlaufend Ereignisprotokollinformationen von anderen Servern unter Windows NT ab. Nachdem die ELA-Daten gesammelt wurden, können sie mithilfe von Hilfsprogrammen analysiert werden.



ELA sammelt unterschiedliche Arten grundlegender Informationen aus den System- und Anwendungsereignisprotokollen. Die gesammelten Daten umfassen Folgendes:

- Grundlegende Informationen zu einem Windows NT-Ereignisprotokoll: die Anzahl von Systemereignissen, der Zeitstempel des ersten Systemereignisses und die Anzahl von Anwendungsereignissen.
- Den Zeitstempel aller Systemneustarts. Auf Computern unter Windows NT 4.0 Service Pack 4 oder höher werden außerdem die Zeitstempel der Ereignisse beim Herunterfahren des Systems gesammelt.
- Informationen zu Windows NT-Systemabstürzen (Fehlerüberprüfungen oder Bluescreens).
- Informationen zu Anwendungsabstürzen (Speicherabbilder im Benutzermodus oder Dr. Watson-Benachrichtigungen).

Eine vollständige Liste der von ELA gesammelten Ereignisse finden Sie unter "Von ELA gesammelte Ereignisse".

ELA ist auf eine unauffällige Struktur ausgelegt. Da ELA nur öffentlich dokumentierte Schnittstellen für den Remotezugriff auf das Ereignisprotokoll verwendet, muss auf den Systemen, auf denen sich die Ereignisprotokolle befinden, keine Software installiert werden. Die Installation von ELA ist ganz problemlos. Es handelt sich um eine einzelne ausführbare Datei, die auf jedem System unter Windows 2000 oder höher ausgeführt werden kann. ELA wirkt sich kaum auf die jeweiligen Produktionsumgebungen aus. In der Regel greift ELA während des Sammlungssprozesses weniger als 30 Sekunden auf einen Remoteserver zu. In Tests mit dem unternehmenseigenen Rechzentrum von Microsoft hat ELA die Ereignisprotokolle von über 1.500 Servern in ca. 40 Minuten mithilfe eines Sammlungssystems mit einer LAN-Verbindung von 100 Mbit/s durchsucht.

Installieren von ELA

So installieren Sie ELA von einer Diskette:

```
C:\> mkdir c:\ela
C:\> copy a:\ela.exe c:\ela
```

So entfernen Sie ELA von einem System:

```
C:\> del c:\atela\*.*
C:\> rmdir c:\atela
```

Konfigurieren von ELA für das Sammeln von Daten

Eine Datei mit einer Liste von Servern gibt ELA die erforderlichen Informationen zum Suchen nach Ereignisprotokollen. Bei der Serverliste handelt es sich um eine einfache Textdatei. Das folgende Beispiel beinhaltet eine Serverliste zum Sammeln von Daten von zwei Windows-basierten Servern:

```
# My server list
SERVER-01
SERVER-05
```

Jede mit einer Raute (#) beginnende Serverlistenzeile ist eine Kommentarzeile, die nicht von ELA verarbeitet wird. Alle anderen Zeilen werden als Windows NT-Servernamen angesehen. In der Regel verwenden Serverlisten die TXT-Dateierweiterung.

In der Serverliste sollte eine Kommentarzeile zum Identifizieren der Rollen der einzelnen Server im Rechenzentrum verwendet werden. Diese Rollen können u. a. SQL Server, Webserver, Datei-/Druckerserver und Domänencontroller umfassen. Zur Aufrechterhaltung einer einheitlichen Datensammlung ist es empfehlenswert, sich an Microsoft zu wenden, wenn Sie die Namen der Sammlungsserver ändern möchten oder das Hinzufügen oder Entfernen von Servern aus der Sammlung planen.

Das folgende Beispiel beinhaltet eine Serverliste zum Sammeln von Daten von zwei Webservern, drei Computern mit SQL Server und einem Domänencontroller:

```
# Web Servers
SERVER-01
SERVER-02
# SQL Servers
SERVER-03
SERVER-04
SERVER-05
# Domain Controller
SERVER-06
```

Konfigurieren von Remoteservern für das Sammeln von Daten

Im Allgemeinen ist keine Konfiguration der Remoteserver erforderlich. Es gibt jedoch optionale Systemeinstellungen, die die ELA-Ergebnisse verzerren können. Standardmäßig sind Windows-basierte Server so konfiguriert, dass sie bei Systemabstürzen (STOP-Fehler oder Windows NT-Bluescreens) Ereignisprotokolleinträge vornehmen.

So aktivieren Sie diese Einstellung

1. Klicken Sie auf **Start**, zeigen Sie erst auf **Einstellungen** und dann auf **Systemsteuerung**, klicken Sie auf **System**, und klicken Sie anschließend auf die Registerkarte **Erweitert**.
2. Klicken Sie im Bereich **Starten und Wiederherstellen** auf **Einstellungen**. Die Einstellung **Ereignis in Systemprotokoll eintragen** sollte aktiviert sein, wenn STOP-Fehler auftreten.

Neben der ELA-Nutzung von Ereignisprotokollen für die Datensammlung ist es empfehlenswert, die Ereignisprotokollgröße des Remoteservers auf 20 MB zu erhöhen. Sie müssen die folgenden Schritte ausführen, um die Ereignisprotokolle eines Computers mit Windows 2000 Server zu vergrößern:

1. Öffnen Sie die Ereignisanzeige, indem Sie im Startmenü auf **Programme**, auf **Verwaltung** und auf **Ereignisanzeige** klicken.
2. Klicken Sie auf ein Ereignisprotokoll.
3. Klicken Sie auf **Vorgang**, und wählen Sie die Option **Eigenschaften** aus.
4. Erhöhen Sie den Wert von **Maximale Protokollgröße** auf 20000 (20 MB).

Sammeln von Daten

ELA-Verwendung:

```
Event Log Analyst, Version 06.03.00.1029
Collects specified event log entries from a list of remote machines
(C) Copyright 1998-2001, Microsoft Corporation
Used by permission only - do not distribute.
ELA ServerList [/t:n /r:n /n:Name]
ServerList Text file containing list of servers to process,
each line of the file should contain a single servername
NOTE: Not used with /c switch.
/t:n Use n threads to process list, default is 40, max is 200
/n:Name Use Name to build the output filenames,
helps you to distinguish one collection from another
/r:<retry count> # of times to retry collecting from a particular server
/i:<init file> Gather custom events specified in <init file>
/b:<path> Log to a database located at <path>
(/s switch not used with this switch)
/c Perform an incremental collection against the database specified
with /B or /S
(/b or /s switch required with this switch)
NOTE: Not used with the ServerList parameter.
/s:<string> Enter user-defined database connection string
(/b switch not used with this switch)
If not given the /b switch, ELA generates the following CSV files:
Autocheck All instances of automatic disk checking at startup.
BugChecks List of all STOP errors found.
DrWatsons Usermode access violations recorded by Dr. Watson.
ExchangeAll All exchange events.
OutOfVM All occurances of the Out of Virtual Memory pop-up.
RAW List of all events recorded by ELA on this run.
RebootReasonCollector All reboot reason events.
Reboots All detected reboots.
SCM All Service control manager events.
ServerErr List of servers not processed, with explanation.
Servers List of servers successfully processed, with additional
detail.
Summary Lists ELA start/stop/elapsed times and the number of
servers processed during each run.
ELA generates the following TXT files:
ServerList Lists each server and the last event collected. For
incremental collections
RunInfo Machine info and Time taken to process each server
Each filename is in the form
ELAXXXXXX_Name_Day_Month_Year_OutputFilename.csv
For example: ELA060000_WebServers_1_Feb_1999_Reboots.csv
```

Nachdem die Serverlistendatei erstellt wurde, können Sie die Ereignisprotokolldaten mithilfe von ELA sammeln.

```
C:\>ela servers.txt
servers.txt: Starting the collection at 19-May-2000 15:54:17
Processing logs on ARAGORN from the beginning(1/43).
...
Processing logs on X1DOCSERV from the beginning(43/43).
Total time elapsed: 00:04:15
All output files written to ELA_19_May_2000_*.csv
C:\>
```

ELA zeigt für jedes durchsuchte Ereignisprotokoll eine Ausgabezeile an. Diese Zeile umfasst Folgendes:

- Den Servernamen, z. B. "SERVER-01"
- Die Serveranzahl aus der Serverliste – wird angezeigt als "(<aktuelle Servernummer>/<Gesamtanzahl der Server>)". Die Anzeige "(4/55)" bedeutet beispielsweise, dass sich dieser Eintrag auf den vierten Server in einer Liste von 55 Servern bezieht.

ELA verwendet standardmäßig 40 Arbeitsthreads, um die Ereignisdaten von bis zu 40 Servern gleichzeitig sammeln zu können. Sobald ein Thread alle Daten eines Servers vollständig gesammelt hat, wird ihm der nächste Server aus der Liste zugewiesen. Nachdem auf dem letzten Server mit der Sammlung begonnen wurde, wartet ELA, bis alle Sammlungsthreads abgeschlossen sind. Dies kann mehrere Minuten dauern.

Sie können die Anzahl von zu verwendenden Threads mithilfe der Befehlszeilenoption "/t:n" angeben, wobei "n" die Anzahl von gewünschten Threads darstellt. Sie können 1 bis 200 Threads angeben. Mit jedem Thread nimmt der vom System verwendete Arbeitsspeicher zu. Eine wachsende Anzahl von Threads kann die Geschwindigkeit der Sammlung nur erhöhen, wenn keine Einschränkungen seitens der Netzwerkverbindung vorliegen. Für die meisten Netzwerke reichen 40 Threads aus. Wenn Sie über eine LAN-Verbindung von 100 Mbit/s verfügen, ist möglicherweise eine Verwendung von 200 Threads am effektivsten.

Analysieren von ELA-Daten

Bei der Ausführung erzeugt ELA 12 Dateien mit durch Kommas getrennten Werten:

- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_Servers.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_ServerErr.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_Autocheck.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_Bugchecks.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_RAW.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_Reboots.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_DrWatsons.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_OutofVM.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_ExchangeAll.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_SCM.csv
- ELA_<Ihr Sammlungsname>_<Sammlungsdatum>_RebootReasonCollector.csv
- ELA_Summary.csv
- ServerList_<Sammlungsdatum>.txt
- Runinfo_<Sammlungsdatum>.txt

Diese Dateien können in Excel importiert und für verschiedene Trendanalysen verwendet werden. Microsoft entwickelt derzeit einige interne Tools, um weitere Analysen durchführen zu können. Da sie sich aber noch in der Entwicklung befinden, können sie noch nicht für den Vertrieb freigegeben werden.

Auswirkungen von ELA auf die Leistungsfähigkeit

Der Entwurf von ELA sieht nur minimale Auswirkungen auf Produktionsrechenzentren vor. ELA verwendet dieselben Schnittstellen für Anwendungsprogrammierung wie die Anwendung Windows-Ereignisanzeige. Die Auswirkungen von ELA auf die Leistungsfähigkeit sind ungefähr so, als ob ein Operator die Ereignisanzeige auf einem Remotesystem verwendet und dabei schnell die Seiten mit den Ereignisinformationen anzeigt. Die ELA-Sammlungszeiten sind jeweils von der Größe des Ereignisprotokolls und von der Bandbreite der Netzwerkverbindung abhängig.

Erforderliche Berechtigungen zum Sammeln von Ereignissen mit ELA

Es wird dringend empfohlen, ELA ausgehend von einem Konto mit Administratorrechten auf den Zielsystemen auszuführen. Die Zeitzone der Zielsysteme kann z. B. großen Einfluss auf die von ELA durchgeführten Berechnungen für den Datenabruf haben und ist nur verfügbar, wenn der Benutzer über Administratorrechte verfügt.

Der Zugriff auf die Ereignisprotokolle wird von dem Konto ermittelt, unter dem die Anwendung ausgeführt wird. Bei dem Konto **LocalSystem** handelt es sich um ein spezielles Konto, das von den Windows NT-Diensten verwendet werden kann. Das Konto **Administrator** setzt sich aus den Administratoren des Systems zusammen. Das Konto **Server-Operatoren** (**ServerOp**) setzt sich aus den Administratoren des Domänenservers zusammen. Das Konto **Jeder** umfasst sämtliche Benutzer auf allen Systemen. ELA greift auf die Ereignisprotokolle **Anwendung** und **System**, nicht aber auf das Ereignisprotokoll **Sicherheit** zu.

Aus der folgenden Tabelle geht hervor, welche Konten über Zugriffsberechtigungen zum Lesen, Schreiben und Löschen für die einzelnen Protokolle verfügen.

Protokoll	**Konto**	**Zugriff**		
Anwendung	LocalSystem	Lesen	Schreiben	Löschen
	Administrator	Lesen	Schreiben	Löschen
	ServerOp	Lesen	Schreiben	Löschen
	Jeder	Lesen	Schreiben	
System	LocalSystem	Lesen	Schreiben	Löschen
	Administrator	Lesen	Schreiben	Löschen
	ServerOp	Lesen		Löschen
	Jeder	Lesen		

ELA verwendet die Funktion **OpenEventLog**, für die eine Lesezugriffsberechtigung erforderlich ist. Ein Mitglied des Kontos **Server-Operatoren** kann die Funktion **OpenEventLog** für die Ereignisprotokolle **Anwendung** und **System** aufrufen, weil das Konto **Server-Operatoren** für diese beiden Protokolle über Lesezugriff verfügt. Ein Mitglied des Kontos **Server-Operatoren** kann jedoch die Funktion **OpenEventLog** nicht für das Ereignisprotokoll **Sicherheit** aufrufen, weil das Konto keinen Lesezugriff auf dieses Protokoll hat.

Von ELA gesammelte Ereignisse

Ereignisquelle	**Ereignis-ID**	**Ereignis-Sub-ID**	**Kommentare**
EventLog	6008		Das System wurde unerwartet heruntergefahren
EventLog	6005		Systemstart
EventLog	6006		Das System wird heruntergefahren
EventLog	6009		Betriebssystemversion während des Startvorgangs
Save Dump	1000, 1001		Bluescreenereignis
SysMgmt	4188		Bluescreenereignis
DrWatson	Alle		Anwendungsfehler
Application Popup	26		Fehler, weil dem System nicht mehr genügend virtueller Speicher zur Verfügung steht
Autochk	1001		Chkdsk wurde während des Systemstarts ausgeführt, um nach Fehlern zu suchen

IIS 5.0 Recycle-Tool

IIS5Recycle wird als Dienst auf einem Webserver unter Windows 2000 mit Internet-Informationsdienste 5.0 ausgeführt. Zweck dieses Tools ist die automatische Wiederverwendung von IIS-Prozessen basierend auf Wiederverwendungskonfigurationen, die in der Registrierung gespeichert sind.

Übersicht über die Funktionalität

Dieses Tool ermöglicht die automatische Wiederverwendung von IIS-Prozessen basierend auf folgenden Faktoren:

- Nutzung des virtuellen Arbeitsspeichers (**InetInfo.exe** + **DLLHost.exe** für Out-of-Process-Anwendungen).
- Anzahl der insgesamt verarbeiteten HTTP-Anforderungen (Leistungsindikator **GET-Anforderungen insgesamt**).
- Geplant (IIS5Recycle wird täglich oder wöchentlich um 1:00 Uhr <lokale Zeit> gestartet).
- IIS-Betriebszeit-basiert (IIS5Recycle wird gestartet, nachdem das Programm 240 Stunden <in Stunden> ausgeführt wurde).
- ASP-Anforderungen in der Warteschlange (IIS5Recycle wird gestartet, wenn die Anzahl von sich in der Warteschlange befindlichen ASP-Anforderungen den unter *%ASPThresholdRetries%* konfigurierten Wert übersteigt.
- Auf einem Windows-System mit Netzwerklastenausgleich wird der Webserver aus dem Cluster entfernt, bevor der IIS-Prozess wieder verwendet wird.
- Bereitstellen der Fähigkeit, einen benutzerdefinierten Befehl (ein Befehlsskript) vor und nach jeder IIS-Wiederverwendung auszuführen.
- Erzwingen Sie das Beenden von IIS, wenn eine Wiederverwendungsanforderung nach einer bestimmten Zeit nicht erfüllt wurde. Dadurch wird sichergestellt, dass IIS bei Erreichen eines Grenzwertes wieder verwendet wird.
- Fähigkeit, die Einstellungen für die Wiederherstellung des IIS-Prozesses über eine Benutzeroberfläche zu konfigurieren.
- Fähigkeit, *IIS5Recycle* im Überwachungsmodus auszuführen, ohne dabei den IIS-Prozess wiederherzustellen. Dadurch kann der Administrator die Wiederverwendungseinstellungen basierend auf den bestimmten Anwendungs- und Umgebungsgegebenheiten abstimmen und optimieren.

Wiederherstellungsfunktion

Der über das MMC-Snap-In **Computerverwaltung** verfügbare Dienststeuerungs-Manager kann so konfiguriert werden, dass er eingreift, wenn ein Dienst unerwartet ausfällt. Der IIS5Recycle-Dienst verwendet diese Funktion als Benachrichtigungs-/Wiederherstellungsmethode bei schwerwiegenden Fehlern. Wenn einer der im Folgenden aufgeführten Fehler auftritt, wird der IIS5Recycle-Dienst beendet, um die Dringlichkeit der Wiederherstellungsmaßnahme des Dienststeuerungs-Managers (Service Control Manager oder SCM) zuzuordnen. Beispiel:

Nachdem der IIS-Prozess beendet wurde, konnte IIS5Recycle den Prozess nach mehreren Wiederholungsversuchen starten. Nachdem der Server bei einem WLBS-fähigen System aus dem Cluster entfernt wurde, konnte IIS5Recycle den WLBS nicht neu starten.

Anmerkung Es wird unabhängig davon, ob die Wiederherstellungsfunktion aktiviert ist, ein Ereignis in Windows EventLog protokolliert.

Durch Konfigurieren der Wiederherstellungsmaßnahmen für den IIS5Recycle-Dienst wird der Administrator benachrichtigt, wenn ein nicht wiederherstellbares Ereignis auf dem Webserver aufgetreten ist.

Installieren des IIS 5.0 Recycle-Tools

1. Öffnen Sie eine Eingabeaufforderung.
2. Navigieren Sie zum Verzeichnis **C:\Programme\Microsoft\IISRecycle**.
3. Installieren Sie den IIS5Recycle-Dienst, indem Sie Folgendes eingeben: **IIS5recycle /install**.
4. Importieren Sie die Datei **IIS5Recycle.reg** in die lokale Registrierung (Anmerkung: NUR für optionale Werte).
5. Konfigurieren Sie die Wiederverwendungsbedingungen, indem Sie Folgendes eingeben: **IIS5Recycle /config**.
6. Starten Sie den/IIS5Recycle-Dienst, indem Sie Folgendes eingeben: **net start IIS5recycle**

Registrierungswerte von IIS Recycle

Sämtliche Konfigurationseinstellungen von IIS Recycle werden in der Registrierung im folgenden Verzeichnis gespeichert:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\IIs5Recycle. Sämtliche Registrierungsänderungen werden automatisch von IIS5Recycle übernommen. Weitere Informationen finden Sie in der folgenden Tabelle.

Value Name(UI Configurable)	Type	Description	Default
RunningHours	REG_DWORD	Recycle IIS process every *%RunningHours%*	24 *7 hours
RequestNumber	REG_DWORD	Recycle IIS process after received *%RequestNumber%* requests	200K Requests
TimeSchedule	REG_SZ	Recycle IIS process at certain times of a day.	N/A
MemoryUsage	REG_DWORD	Recycle IIS process if it used more than *%MemoryUsage%* KB memory (Commit Charge)	1,000,000 KB
ASPRequestsQueued	REG_DWORD	Recycle IIS process if the ASP Requests Queued is above this value in %ASPThresholdRetries% retries.	2000
Value Name(No UI, Reg Configurable)	**Type**	**Description**	**Default**
QueryIisMinutes	REG_DWORD	How often our process will query IIS status.	15 minutes
RecycleIisDelay	REG_DWORD	If a recycle condition is reached, how long to wait to see if IIS can reach a quiet time define by *%QuietTimeRequestNumber%*.	0 minutes (Recycle Immediately)
QuietTimeRequestNumber	REG_DWORD	If a recycle condition is reached, we will try to recycle IIS at the next quiet time (if *%RecycleIisDelay%* != 0). This number will be used to decide if IIS is quiet or not.	10 Get Requests / Sec
IISStopTimeOutSecs	REG_DWORD	Time given IIS to stop	60 seconds
IISStartTimeOutSecs	REG_DWORD	Time given IIS to start	60 seconds
NoRealRecycle	REG_DWORD	Report Recycle event only	0 (do recycle)
WLBSDrainWaitSecs	REG_DWORD	Time given WLBS to drain stop.	60 seconds
WLBSWaterMark	REG_DWORD	If the number of current active hosts in the NLB cluster is below or equal to this threshold, Iis5Recycle will skip current recycle period util the active hosts is above the threshold	0 (Disabled)
WLBSDrainStopConnectionNum	REG_DWORD	If the current connection is below this value, Iis5Recycle will stop draining.	0 (drain until there is no connection left)
ForceDrainStop	REG_DWORD	Enforce NLB is not started by other monitoring program during the *%WLBSDrainWaitSecs%* period.	0 (Disabled)
ASPThresholdRetries	REG_DWORD	How many time should Iis5Recycle retries when ASP requeusts queued is above *%ASPRequestsQueued%* value.	0 (no retries)

Anmerkung Mit den ersten fünf konfigurierbaren Registrierungswerten werden die Wiederverwendungsbedingungen für IIS festgelegt, die IIS5Recycle verwendet, um zu ermitteln, in welchen Fällen der IIS-Prozess wieder verwendet werden soll. Bei den übrigen Registrierungswerten handelt es sich um optionale Werte, die durch Importieren der Datei **IIS5Recycle.reg** in die lokale Registrierung zur Verfügung gestellt werden.

IIS5Recycle ruft bei jeder Aktivierung Registrierungsänderungen ab.

NoRealRecycle

Mit dieser Funktion können Sie das IIS 5 Recycle-Tool im Überwachungsmodus ausführen, ohne dabei den IIS-Dienst zu unterbrechen. Beim Verwenden im Überwachungsmodus werden die IIS Recycle-Ereignisse im Windows-Ereignisprotokoll gespeichert, und sie können verwendet werden, um die IIS Recycle-Bedingungen zu beurteilen, die für den Webserver konfiguriert werden sollten.

Advanced Control (die Skripterstellungsfunktion)

Die "Advanced Control"-Benutzeroberfläche ermöglicht den Kunden die Angabe eines Befehls (Skripts), der vor und nach jeder IIS-Wiederherstellung ausgeführt werden soll. Der Rückgabewert des Befehls, der vor der IIS-Wiederherstellung ausgeführt werden soll, wird überprüft, wenn das Kontrollkästchen **Verify return value** aktiviert ist. Und wenn das Skript nicht den erwarteten Wert zurückgibt, behandelt IIS5Recycle dies als Fehler und überspringt den aktuellen Wiederverwendungsvorgang. Die Ausgabe des Skripts wird an eine Protokolldatei umgeleitet, die sich im folgenden Verzeichnis befindet: **%windir%\system32\LogFiles\IIs5Recycle**.

Wenn in der Protokolldatei nicht die erwartete Ausgabe angezeigt wird, stellen Sie unbedingt sicher, dass der Skriptname korrekt ist und in das Verzeichnis **%windir%\system32** kopiert wurde. Wenn das Skript auf andere ausführbare Dateien zugreifen muss, dann stellen Sie sicher, dass sich all diese Dateien ebenfalls im Verzeichnis **%windir%\system32** befinden.

Auf einem WLBS-fähigen System wird das Skript nach Beendigung des Netzwerklastenausgleichs und vor dem Starten des Netzwerklastenausgleichs ausgeführt.

Wasserzeichenfunktion für WLBS

Es gibt zwei Anforderungen für das Verwenden der Wasserzeichenfunktion:

- Aktivieren der Remotesteuerung auf jedem WLBS-Knoten.
- Wenn ein Kennwort für die Remotesteuerung erforderlich ist, muss jeder Knoten dasselbe Kennwort verwenden.

IIS Recycle-Ereignisse

Die folgende Liste enthält Beispiele für die von IIS Recycle protokollierten Ereignisse.

EVENT ID(Error)	Event Message	Parameters
MSG_NOT_W2K	Iis5Recycle wasn't supported on a non-Windows 2000 machine.	N/A
MSG_IIS_START_FAILED	Unable to start IIS server, error: %1	%1 – error code OR error message
MSG_IIS_STOP_FAILED	Unable to stop IIS server, error: %1	%1 – error code OR error message
EVENT ID(Warning)	**Event Message**	**Parameters**
MSG_IIS_MEMORY_EXCEEDED	IIS server memory usage has exceeded the limit %1 KB. %2	%1 – the memory usage limit set by the customer. %2 – the current status of IIS server, such as: current memory usage, request number, running time, and ASP queue size.
MSG_IIS_TIME_EXCEEDED	IIS server has run for more than %1 hour(s). %2	%1 – the time elapsed since IIS was started. %2 – current IIS status (same as above)
MSG_IIS_SCHEDULE_REACHED	IIS server has reached scheduled restart time %1. %2	%1 – the specific time need to restart IIS. %2 – current IIS status (same as above)
MSG_IIS_REQUEST_EXCEEDED	IIS server has received more than %1 requests. %2	%1 – the recycle request number. %2 – current IIS status (same as above)
MSG_IIS_RESTARTED	IIS server was successfully restarted.	N/A
EVENT ID(Information)	**Event Message**	**Parameters**
MSG_SERVICE_STARTED	Service '%1' started	%1 – the name of the service.
MSG_SERVICE_STOPPED	Service '%1' stopped	%1 – the name of the service.

Das Tool VBCHKW2K

Anforderungen von VBCHKW2K

- Unterstützt nur Windows 2000-Systeme
- Erfordert Administratorrechte

Funktionen von VBCHKW2K

VBCHKW2K.exe kann die folgenden Funktionen durchführen, wenn die Ausführung von einer Befehlszeile erfolgt:

- Eine *diskret im Hintergrund ausgeführte* Überprüfung des COM+-Katalogs (bei der auch nicht instanziierte Komponenten erkannt werden) im Hinblick auf nicht ordnungsgemäß kompilierte VB-DLLs, die in COM+-Anwendungen registriert sind.
- Suchen nach VB-Komponenten, die in IIS-In-Process-Anwendungen (**inetinfo.exe**) ausgeführt werden.
- Suchen nach VB-Komponenten, die in IIS-Out-Of-Process-Pooled-Anwendungen (**dllhost.exe**) ausgeführt werden.
- Suchen nach VB-Komponenten, die in IIS-Out-Of-Process-Anwendungen (**dllhost.exe**) ausgeführt werden.
- Suchen nach VB-Komponenten, die in eigenen COM+-Servern (**dllhost.exe**) ausgeführt werden.

Anschließend erstellt VBCHKW2K einen Bericht, der auf dem Bildschirm angezeigt wird und sämtliche DLLs auflistet, die nicht ordnungsgemäß kompiliert wurden. Ein Beispielbericht kann folgendermaßen aussehen:

```
c:\vbchkw2k
VBCheckW2k Version 1.2.1a
Microsoft Corporation - Dec 2000
Mon Dec 11 07:15:47 2000
Scanning IIS, all Medium & High Isolation Web applications and all running
COM+ server
packages on SIEWEB...Scanning dllhost.exe (PID:1376) System Package
Scanning inetinfo.exe (PID:5436) IIS In-Process Applications
Retained In Memory :OFF
Unattended Execution :OFF
C:\Inetpub\SIEWeb\vbrimcheck\VBRimCheck.dll <<< Please Recompile!
* IMPORTANT NOTE
VBCHKW2K has detected one or more Visual Basic DLL's that have not been
compiled properly
for use in a server process such as IIS or COM+. Please make note of the
DLL's listed
above and re-compile these DLL's with 'Unattended Execution' and 'Retain in
Memory'.
```

Im obigen Bericht hat das Tool eine DLL mit der Bezeichnung **VBRimCheck.dll** gekennzeichnet, die "prozessintern" von IIS ausgeführt wird. Wenn das Tool eine nicht ordnungsgemäß kompilierte DLL erkennt, wird am Ende des Berichts ein wichtiger Hinweis angezeigt, der auf das Problem aufmerksam macht.

Zusammenfassung

Verwenden Sie die unten aufgeführte Liste, um die zehn wichtigsten Empfehlungen kennen zu lernen, die Ihre Organisation nutzen kann, um die Zuverlässigkeit und Verfügbarkeit der Webserver zu verbessern:

- Wenn Sie die Anwendung in Visual Basic entwickeln, sollten Sie das Dienstprogramm VBCHKW2K verwenden, um die Richtigkeit der Kompilierungseinstellungen überprüfen zu können.
- Testen Sie die vorhandene Hardware und Software sorgfältig im Hinblick auf die Windows 2000-Kompatibilität.
- Verwenden Sie möglichst häufig gepoolte ASP-Seiten.
- Verwenden Sie möglichst häufig gepoolte COM-Komponenten.
- Verwenden Sie das Web Capacity Analysis-Tool und die HTTP-Überwachung, um Belastungstests der Anwendung durchzuführen.
- Dokumentieren und befolgen Sie einen Webserver-Bereitstellungsplan.
- Installieren und verwenden Sie das IIS 5 Recycle Tool, um die Webserververfügbarkeit zu erhöhen.
- Dokumentieren und befolgen Sie einen Prozess zur Auswertung und Prioritäteneinteilung der Anwendbarkeit von freigegebenen Hotfixes und Sicherheitsaktualisierungen.
- Verwenden Sie HFCheck und QFECheck, um eine Betriebssystem-Standardinstallation auf allen Servern sicherzustellen.
- Wählen Sie ein Ziel für die Serververfügbarkeit aus, und verwenden Sie Microsoft-Tools, um den bisherigen Erfolg dieses Zieles nachverfolgen zu können.

Die in diesem Dokument enthaltenen Informationen stellen die behandelten Themen aus der Sicht der Microsoft Corporation zum Zeitpunkt der Veröffentlichung dar. Da Microsoft auf sich ändernde Marktanforderungen reagieren muss, stellt dies keine Verpflichtung seitens Microsoft dar, und Microsoft kann die Richtigkeit der hier dargelegten Informationen nach dem Zeitpunkt der Veröffentlichung nicht garantieren.

Dieses Whitepaper dient nur zu Informationszwecken. MICROSOFT SCHLIESST FÜR DIESES DOKUMENT JEDE GEWÄHRLEISTUNG AUS, SEI SIE AUSDRÜCKLICH ODER KONKLUDENT.

Die Benutzer/innen sind verpflichtet, sich an alle anwendbaren Urheberrechtsgesetze zu halten. Unabhängig von der Anwendbarkeit der entsprechenden Urheberrechtsgesetze darf ohne ausdrückliche schriftliche Erlaubnis der Microsoft Corporation kein Teil dieses Dokuments für irgendwelche Zwecke vervielfältigt oder in einem Datenempfangssystem gespeichert oder darin eingelesen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln (elektronisch, mechanisch, durch Fotokopieren, Aufzeichnen usw.) dies geschieht.

Microsoft Corporation kann Inhaber von Patenten oder Patentanträgen, Marken, Urheberrechten oder anderem geistigen Eigentum sein, die den Inhalt dieses Dokuments betreffen. Das Bereitstellen dieses Dokuments gibt Ihnen jedoch keinen Anspruch auf diese Patente, Marken, Urheberrechte oder auf sonstiges geistiges Eigentum, es sei denn, dies wird ausdrücklich in den schriftlichen Lizenzverträgen von Microsoft eingeräumt.

© 2001. Microsoft Corporation. Alle Rechte vorbehalten. Microsoft, Active Server Pages, Windows und Windows NT sind eingetragene Marken oder Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere in diesem Dokument aufgeführte tatsächliche Produkt- und Firmennamen können geschützte Marken ihrer jeweiligen Inhaber sein.

[1] Blischke, Wallace R. and Murthy, D.N Prabhakar "Reliability: Modeling, Prediction, and Optimization" (englischsprachig)